TRANSIT SYSTEM SCHEDULING WITH LIMITED VEHICLE CAPACITY

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY

124140

by
RAVI AGRAWAL

to the
Department of Civil Engineering
Indian Institute of Technology, Kanpur
Augest, 1997

5 DEC 1997
CENTRAL LIBRARY
I. I. T., KANPUR
Acc. No. A 124440

A124440

CE-1997-M-AGR-TRA

CERTIFICATE

This is to certify that the present research work entitled " Transit System Scheduling With Limited Vehicle Capacity " *has been carried out by* RAVI AGRAWAL *under my supervision and it has not been submitted elsewhere for a degree.*

Dr. P. CHAKROBORTY

Assistant Professor

Department of Civil Engineering

Indian Institute of Technology

Kanpur

AUGUST 1997

Dedicated to

My Parents

ACKNOWLEDGEMENT

It is my great privilege to express a deep sense of gratitude to the ALMIGHTY for all I am blessed with and then to my teachers, parents and colleagues, blessings of all those who made me able to complete this work.

First, I am deeply greatful to my adviser, Dr. Partha Chakroborty, for his invaluable guidance, affectionate and moral encouragement throughout the course of this work. My sincere thanks are due to Dr. Kalyanmoy Deb, of Mechanical Engineering for his useful suggestions.

I sincerely appreciate the companionship of all my friends - Capt. Sharma, Pankaj, G. Kulkarni, Gurvinder, Vipin, Ashish, Ajit, Rajesh, Ramesh and special thanks to Harikishan Reddy for all sorts of help. I also express my heartful appreciation to all staff members of Civil Engineering Department for their help.

IIT Kanpur
August 21, 1997

Ravi Agrawal

Abstract

The objective of any transit system is to provide good level of service to its users with the available resources. Waiting time of the passengers during their journey is a good measure of level of service. Total waiting time on a transit system includes the transfer time, of passengers transferring between different routes and the initial waiting time of the passengers waiting to board a bus, at their point of origin. The overall waiting time can be minimized through proper scheduling of the transit system. In this study, the scheduling problem for a transit system, in which capacities of vehicles (say buses) are limited (i.e., a may not be able to accommodate all the passengers who are waiting for that bus at the station) is considered. First a mathematical programming (MP) formulation for this problem is presented. However, due to the large number of variables and non-linearity, it is difficult to solve the MP formulation by traditional methods. Hence, a Genetic Algorithm (Genetic Algorithms are powerful search and optimization methods) based procedure is developed and used to solve the problem. Function based declarations and coding of variables possible in GA - based algorithms allow an efficient reformulation of the original problem. The revised formulation of the problem is computationally much simpler than the original formulation of the problem (it may be noted that one can revise the formulation because of certain features present in GAs). Results from a number of test cases show that the proposed procedure is able to find optimal schedules with reasonable computational resource.

Contents

List of Figures

Chapter 1

Introduction

Public transportation systems play an important role in transporting people and goods. It is therefore, desirable to have an efficient public transport system which provides comfort, safety and better level of service to the users. Further in order for such systems to be sustainable they need to be economical/profitable to the operator. One of the sustainable ways of improving the level of service to the users is by developing a schedule which provides lesser waiting time to passengers at no extra cost to the operator. Hence, development of an optimal schedule is of prime importance in designing a good public transportation system.

In most public transportation systems, direct routes between every possible origin - destination pair do not exist. Sometimes, persons need to transfer from one route to another in order to reach their destination. Thus, two types of "waiting" passengers exist at a transfer station (i.e., a station where different routes intersect); those whose point of origin is the station, and those who have arrived at the station on some route and wish to transfer to another route. The former may be called be called non-transferring passengers and the later transferring passengers. A good schedule, therefore, is one which minimizes the sum of the total waiting time for all non-transferring passengers and the

total waiting time for transferring passengers.

Developing schedules which minimize the waiting time of all passengers (TWT) is a difficult problem. It was shown earlier (Chakroborty et al. [1], Kikuchi and Parmeswaran [2]) that use of traditional optimization tools fail to provide an optimal schedule even for simple cases of the scheduling problem. Chakroborty et al. [1] also showed that Genetic algorithms (GAs) could be effectively used to obtain optimal schedules at a transfer station. Later, Chakroborty et al. [3] extended the procedure to obtain network - wide optimal schedules. Recently Chakroborty et al. [4] developed a procedure to obtain optimal schedules when arrival times of buses are stochastic (i.e., strict adherence to the schedules is not possible). However, in all the above works the capacity of the buses is assumed to be sufficiently large so as to accommodate all those who are waiting for the bus. In the present thesis this assumption is relaxed. The GA based procedure developed here can successfully obtain optimal schedules when the bus capacity is limited, that is when not all passengers can board the bus for which they have been waiting.

The rest of this thesis is divided into 6 chapters. Chapter 2 provides a detailed description of the problem and Chapter 3 presents the traditional formulation for the difficulties and complexities associated with the traditional formulation. Chapter 4 presents a brief introduction to Genetic Algorithms, a new yet powerful optimization tool. Later, Chapter 5 presents a revised formulation of the problem. This revised formulation is developed keeping in mind that GAs will be used as the optimization tool. Chapter 6 presents the best schedules obtained by using the proposed algorithm for a variety of test cases. The results show the efficacy of the proposed algorithm. The conclusions from this study are presented in Chapter 7.

Chapter 2

Problem Statement

This chapter describes the problem of optimal transit system scheduling with transfer time considerations and under the assumption that transit vehicle capacity is finite.

Typical transit system have various routes and stations. Routes occasionally intersect at points referred to as transfer stations. These transfer stations are shown by the circles in the Figure 2.1. In Figure 2.1 the solid lines represent the various routes on the network, and the dots on the lines represent the stops on the routes. The objective is to determine a schedule for the transit vehicle (either buses or trains) on each route which provides a better Level of Service (LOS) to the passengers within the available resources. A good measure of level of service is the waiting time of passengers on the system. The duration of time period passengers spend in waiting during their journey from origin to destination is the, "waiting time". A lesser waiting time translates to a better level of service. At any transfer station on a transit network two types of passengers exist: for some passengers the station is their point of origin and for others the station is a point where they wait to board a bus on a particular route after being dropped off by a bus of another route. For example, consider Station A in Figure 2.1. Two kinds of passengers wait at this station. The first

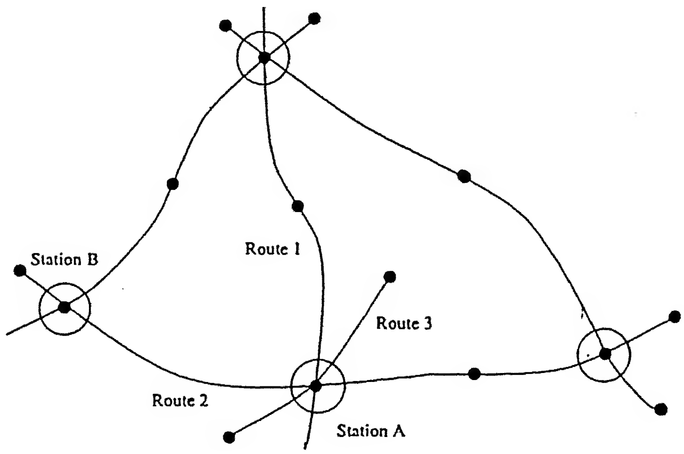

Figure 2.1: A Typical transit system network

type of passengers, whose point of origin is the Station A come to the station and wait to board a bus on one of the routes going through Station A. The waiting time of these passengers is termed as initial waiting time (IWT). The other type of passengers come to Station A from some other station and wait to board a bus on a route which will eventually take them to their destination. These passengers do not have much control over the time at which they arrive at Station A, since that is controlled by the arrival time of the bus on which they came to Station A. The time such passengers spend in waiting at Station A is termed as transfer time (TT). The total time passengers have to wait (TWT) is therefore the sum of the IWT and TT . A good schedule, therefore, is one which minimizes TWT.

It should be noted that the development of an optimal schedule (or any feasible schedule) proceeds under certain resource constraints as well as service related constraints. The following enumerates the important resource constraints and service related constraints which are to be taken into account while developing a schedule.

Resource constraints

1. Fleet size : The schedules should be developed based on the available fleet size on each route in the transit system.
2. Capacity of the bus : Each bus has a finite capacity and may not be able to accommodate all the passengers waiting for that bus.

Service related constraints

1. Minimum stopping time : A bus cannot start as soon as it arrives at a station. It has to stop for a certain minimum period of time.
2. Maximum stopping time : A bus cannot stop for more than a certain period of time at a station.
3. Maximum transfer time : A passenger cannot be allowed to wait for more than a certain period of time for the transfer.
4. Policy headway : The headway on each route should be less then a certain maximum, known as the policy headway.

This thesis concentrates on developing an optimal schedule at a single transfer station under the above constraints. Here, it is assumed that the bus capacity is finite, so that it may not be able to accommodate all passengers waiting for that bus at the station. Hence, some passengers have to wait for the next bus. This will increase the initial waiting time (IWT) of non-transferring passengers as well as transfer time (TT) of transferring passengers. Further, it is assumed that at the station, service is based on first-come-first-served-basis. That is passengers have to wait for a bus in a queue irrespective of whether they are transferring passengers or not. Capacity of the bus will decide how

many passengers this queue will board the bus. The rest will remain, and the new arrivals will join this existing queue.

The next chapter presents and discusses in details a mathematical formulation for this problem.

Chapter 3

Traditional Formulation

In this chapter, the traditional mathematical programming formulation of the scheduling problem described earlier is presented. For ease of understanding, the formulation of problem has been described in two parts: (i) formulation of the problem with the assumption of infinite bus capacity (i.e., the buses always have enough space to accommodate all the passengers waiting for that bus), and (ii) formulation of the problem with finite bus capacity. The problem with infinite bus capacity has been tackled earlier by Subrahmanyam [5] and Chakroborty et al. [1]. The discussion related to infinite bus capacity has been adapted from the references mentioned earlier and is presented here for the sake of completeness. Following this discussion is a presentation of the finite bus capacity version of the above problem. This chapter also includes discussions on the complexities of the problem and on the selection of an efficient algorithm to solve the problem.

3.1 Mathematical Programming Formulation with Infinite Capacity of Buses

The formulation of the problem assuming infinite capacity of buses is presented and discussed in this section. This section is further divided into four sub-sections. The assumptions and decision variables of the problem with infinite capacity of buses are discussed in the first sub-section. The mathematical programming (MP) formulation, explanation on the objective function, and explanation of the constraints are presented in the second, third, and fourth sub-sections, respectively.

3.1.1 Assumptions and Decision Variables

The following assumptions are made while solving the scheduling problem with infinite bus capacity.

1. The fleet size (the number of buses) for each route is known.
2. Minimum stopping time : A bus on any route i has to stop at a station for a minimum period of time, s_i^{min}.
3. Maximum stopping time : A bus on any route i cannot stop for more than a period equal to s_i^{max}.
4. Maximum transfer time : No passenger on the transit network should wait more than time T for a transfer.
5. The number of transfers from each bus of each route to all other routes is known.
6. The arrival pattern of passengers is assumed to be of the form shown in Figure 3.1. In the figure, $v_{i,k}(t)$ refers to the arrival pattern of passengers

(whose point of origin is the transfer station under consideration) for the k^{th} bus of the i^{th} route which departs at d_i^k. The function $E_i(\tau)$ is the locus of the maximum of $v_{i,k}(t)$ for all k. This function is explained later.

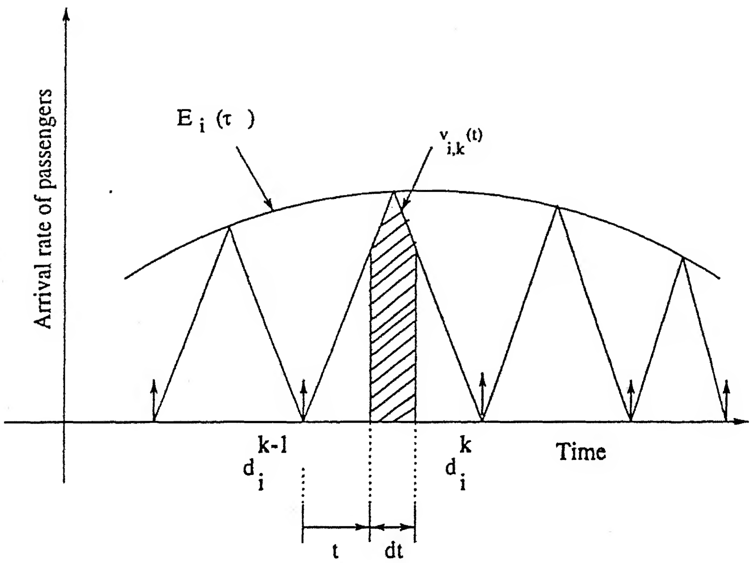

Figure 3.1: Arrival pattern of passengers for Route i

7. The capacity of a bus is large enough to accommodate all the passengers waiting to board the bus (this assumption is termed as "infinite bus capacity").

The decision variables used in the MP formulation given in the next subsection are the arrival times a_i^k (where any a_i^k is the arrival time of the k^{th} bus of the i^{th} route), the departure times d_i^k (where any d_i^k is the departure time of the k^{th} bus of the i^{th} route) and the δ_{ij}^{kl} values. The variable δ_{ij}^{kl} is a 0 - 1 integer variable. If the transfer from the k^{th} bus of the i^{th} route to the l^{th} bus of the j^{th} route is not possible or is not optimal then $\delta_{ij}^{kl} = 0$. Otherwise $\delta_{ij}^{kl} =$ 1.

3.1.2 Formulation

The scheduling problem under the assumptions stated in the previous section can be formulated as:
Minimize

$$\sum_i \sum_j \sum_k \sum_l \delta_{ij}^{kl}(d_j^l - a_i^k)w_{ij}^k + \sum_k \sum_i \int_0^{d_i^k - d_i^{k-1}} v_{i,k}(t)(d_i^k - d_i^{k-1} - t)dt \quad (3.1)$$

Subject to

$$\begin{array}{llll}
G1 & : & (d_i^k - a_i^k) \leq s_i^{max} & \forall i, k \\
G2 & : & (d_i^k - a_i^k) \geq s_i^{min} & \forall i, k \\
G3 & : & (a_i^k - a_i^{k-1}) \leq h_i & \forall i, k \\
G4 & : & (d_j^l - a_i^k)\delta_{ij}^{kl} \leq T & \forall i, j, k, l \\
G5 & : & (d_j^l - a_i^k) + M(1 - \delta_{ij}^{kl}) \geq 0 & \forall i, j, k, l \\
G6 & : & \sum_l \delta_{ij}^{kl} = 1 & \forall i, j, k
\end{array}$$

3.1.3 Objective Function

The main objective of this problem is to develop an optimal schedule for providing better level of service to its passengers. This objective can be obtained by developing a schedule, which minimizes the sum of the initial waiting time of passengers (non-transferring) waiting to board a bus at their point of origin and the transfer times of (transferring) passengers who come from one route to transfer to another route at a transfer station. The objective function consists of two terms. The first is the Total transfer time (TT) and the second the Total initial waiting time (IWT).

In the following, a brief discussion on how these expressions were obtained is provided.

Total transfer time, TT

At a transfer station in a network, the transferring passengers come from

one route, alight at the transfer station and wait there for a bus of another route to which they want to transfer. The time for which all these passengers wait is referred to as the total transfer time.

Let us assume that a passenger wants to transfer from i^{th} route to the j^{th} route and he/she arrives at arrival time, a_i^k, on the k^{th} bus of the i^{th} route. It is understood that transfer to the l^{th} bus on the j^{th} route will be made only if the l^{th} bus departs after a_i^k and also if the l^{th} bus is the first bus on the j^{th} route which is available to that particular passenger. Therefore, the waiting time of this particular passenger will be equal to $\Sigma_l(d_j^l - a_i^k)\delta_{ij}^{kl}$. Where δ_{ij}^{kl} =1 if the l^{th} bus on the j^{th} route meets the above conditions or else it is zero.

Now in general, if w_{ij}^k is the number of passengers who want to transfer from k^{th} bus of i^{th} route to the j^{th} route, then obviously the total transfer time for the w_{ij}^k passengers from k^{th} bus of i^{th} route will become

$$Transfer\ time = \Sigma_l(d_j^l - a_i^k)\delta_{ij}^{kl}w_{ij}^k \tag{3.2}$$

The total transfer time, TT, can now be obtained by summing the above term for all the buses on route i and for all route pairs:

$$TT \ = \Sigma_i\Sigma_{j,j\neq i}\Sigma_k\Sigma_l(d_j^l - a_i^k)\delta_{ij}^{kl}w_{ij}^k \tag{3.3}$$

It is pertinent to note that TT contains the product of two decision variables and hence it is a non-linear expression.

Total initial waiting time, IWT

The station under consideration may also be the originating station for some passengers. The waiting time for these passengers at their point of origin is referred to as the initial waiting time. The passengers who come after the departure of $(k-1)^{th}$ bus and before the departure of k^{th} bus of route i and want to board a bus on route i are able to board the k^{th} bus on i^{th} route. It is

assumed that the passengers arrive at the stop (rate of arrival), during the time interval d_i^{k-1} to d_i^k, according to some function $v_{i,k}(t)$, where, t is measured from d_i^{k-1}. Figure 3.1 shows the distribution of passengers $v_{i,k}(t)$. Here distribution of $v_{i,k}(t)$ is assumed to be triangular (see Figure 3.2). Maximum $v_{i,k}(\mathrm{t})$ is assumed to be at $\mu(d_i^k - d_i^{k-1})$, where $0 \leq \mu \leq 1$.

The area of the triangle gives the total number of passengers waiting to board the k^{th} bus of the i^{th} route. The moment of this area about d_i^k gives the total waiting time of these passengers. In Figure 3.1, $E_i(\tau)$ is the locus of the maximum $v_{i,k}(t)$ for each i. Here τ is related to the departure times as $\tau = d_i^{k-1} + \mu(d_i^k - d_i^{k-1})$. Therefore the total IWT for all routes and all buses is given by

$$IWT = \sum_k \sum_i \int_0^{d_i^k - d_i^{k-1}} v_{i,k}(t)(d_i^k - d_i^{k-1} - t)dt. \tag{3.4}$$

Figure 3.2 further clarifies the above discussion.

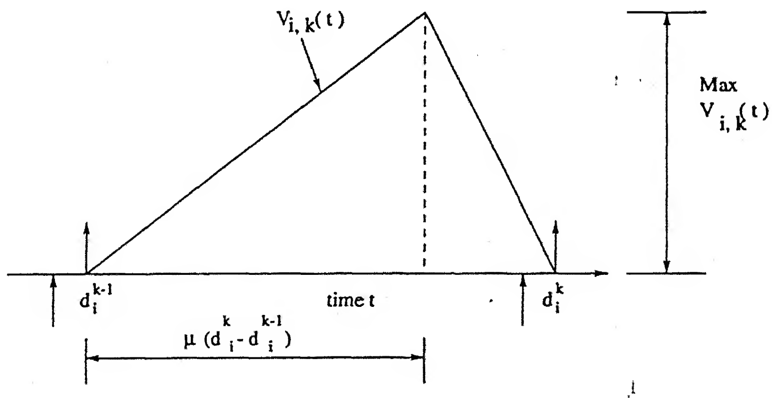

Figure 3.2: Detailed arrival pattern of passengers between two consecutive buses on route i.

It may be noted that IWT term is also non-linear in terms of the decision variables d_i^k.

3.1.4 Constraints

Here each of the constraints stated in the MP formulation are discussed briefly.

1. Maximum stopping time constraints :

   Constraints G1 : $(d_i^k - a_i^k) \leq s_i^{max}$

   Constraints G1 ensure that for any i and k, the stopping time will not be greater than the maximum stopping time.

2. Minimum stopping time constraint :

   Constraints G2 : $(d_i^k - a_i^k) \geq s_i^{min}$

   Stopping time at a station for any bus k on any route i should be such that passengers are able to alight from and board the bus. Therefore, the stopping time should not be too small. Constraints G2 ensure that the stopping times $(d_i^k - a_i^k)$ for any bus on the route i should be greater than or equal to the minimum stopping time s_i^{min}.

3. Headway constraints :

   Constraints G3 : $(a_i^k - a_i^{k-1}) \leq h_i$

   These constraints ensure that each of the headways on each of the routes are below the policy headway h_i.

4. Maximum transfer time constraints :

   Constraints G4 : $(d_j^l - a_i^k)\delta_{ij}^{kl} \leq T$

   Constraints G4 ensure that the transfer time for each passenger be less than a maximum stipulated value, T.

5. Transfer constraints :

   (a) Constraints G5 : $(d_j^l - a_i^k) + M(1 - \delta_{ij}^{kl}) \geq 0$

These constraints ensure that δ_{ij}^{kl} will be zero whenever a transfer from the k^{th} bus of i^{th} route to the l^{th} bus of the j^{th} route is not possible. (When $(d_j^l - a_i^k)$ is negative then $(1 - \delta_{ij}^{kl})$ has to be one, i.e., δ_{ij}^{kl} has to be zero. However, when $(d_j^l - a_i^k)$ is positive then according to these constraints δ_{ij}^{kl} could be either one or zero.)

(b) Constraints G6 : $\sum_l \delta_{ij}^{kl} = 1$

These constraints ensure that one and only one transfer is made from the k^{th} bus of the i^{th} route to the j^{th} route. However, it may be noted that together with objective function the value of δ_{ij}^{kl} will become one for that l of the j^{th} route for which $(d_j^l$ - $a_i^k)$ has the least value. Hence, δ_{ij}^{kl} will be zero for all the other values of l on the j^{th} route.

3.1.5 Difficulties in Solving by Traditional Methods

In this section the difficulties in solving the above problem by traditional optimization tools are discussed. It may be noted that it has been reported elsewhere (Subrahmanyam [5], Chakroborty et al. [1], Srinivas [3], Reddy [4]) that traditional methods fail to converge to an optimal solution for the above problem. The difficulties of solving this problem using traditional means arise due to the following characteristics of the problem:

(i) **Large and complex search space**

In order to understand the extent and complexity of the search space consider a transit station with only three routes intersecting at the station. Let the number of buses plying on each route be 10. In this case there are 660 decision variables, out of which 600 are 0-1 variables. The total number of

constraints for all the buses and for all the routes will be 1350 of which 600 are non-linear constraints. With these many integer variables and non-linear constraints the search space becomes very large and complex (in terms of shape and discontinuity).

(ii) Non-Linearity

TT, IWT terms in the objective function and Constraints G4 are non-linear in the above discussed formulation. This non-linearity adds considerable complexity to the problem. It may be noted that the problem is thus a Mixed Integer Non-linear programming problem and traditional tools for solving such problems are few, highly iterative in nature and therefore not very efficient. In order to utilize some what more efficient algorithms like branch and bound or cutting plan method, Chakroborty et al. [1] linearized a simplified version of the problem (in which they simplified the problem by neglecting the IWT term in the objective function) using Glover's technique (Glover, [6]). However, even then traditional algorithms for solving Linear Mixed Integer programming problems failed to converge to an optimal solution (Chakroborty et al. [1]).

As will be shown in the following section, the scheduling problem with finite bus capacity is a lot more complex than the problem described thus far. For this reason and the experiences stated above no attempt was made to solve the capacity constrained scheduling problem by traditional means. In fact, one can easily appreciate after following the next section that even obtaining a traditional formulation for the capacity constrained problem is almost impossible.

3.2 MP Formulation with Finite Bus Capacity

In the above formulation the bus capacities are assumed to be infinite, that is, always, all the passengers waiting at the station can be accommodated in the bus. But in reality buses have finite (limited) capacities and accordingly can not always accommodate all the passengers waiting at the station to board the bus. Therefore if the assumption of infinite bus capacity is relaxed to finite capacity the optimal schedule obtained from the above formulation may not be optimal any more. In this section, we describe the problem with capacity constraints on buses and discuss its effect on the mathematical programming formulation of the problem.

The assumptions of this problem are the same as those mentioned in the previous section except the assumption regarding infinite bus capacity. Here, it is assumed that all buses have a known but finite capacity.

Before proceeding with the discussion on problem formulation it must be understood that in the earlier formulation it was inherently assumed that all passengers (either transferring or non-transferring) who arrive between the k^{th} and $(k+1)^{th}$ bus of the i^{th} route and want to take route i will board the $(k+1)^{th}$ bus of route i. Obviously this facit assumption is no longer required since some passengers who arrive before the $(k+1)^{th}$ bus may not be able to board that bus due to unavailability of space (i.e., due to capacity restriction). Hence, it becomes imperative now to keep a track of which passenger goes to which bus.

In order to maintain track of each passenger, one needs to develop a queue of passengers for each route. Passengers, either transferring or non-transferring, join the queue of their requirement as and when they arrive at the station. As buses on different routes arrive, the corresponding queues are dissipated based on the buses capacity. By following this queuing system one

can determine which passenger (based on his/her arrival time) will board which bus and hence determine the passenger's initial waiting time or transfer time as the case may be.

The above description is further explained with the help of the Figure 3.3. In this figure only a window of the entire scheduling time period for a particular station is shown. Buses on route i depart at d_i^k, d_i^{k+1} etc. during this window. The top figure shows the arrival pattern (rate of arrival) of non-transferring passengers for route i. The middle figure shows the number of passengers who come from other routes (e.g. j, $j+1$) to transfer to route i and also the time at which these buses arrive. The bottom figure represents the total number of passengers waiting at the station for route i and is derived from the cumulative of the first, the second figure and the departure times of the buses on route i. Initially the total number of passengers waiting rises quadratically (this function is basically the area of the triangle shown in the top figure as a function of t - d_i^k) then at a_j^l there is a sharp jump due to the arrival of transferring passengers from the l^{th} bus of the j^{th} route. This process continues till a bus arrives. When this happens, a maximum of C passengers (where C is the capacity of the bus that arrives) can board. If the total number waiting N is more than C (as is the case shown in Figure 3.3) then the last H (H = N - C) persons are left. This can be seen from the Figure 3.3. Note how the line representing the total number waiting starts from H at time d_i^{k+1} + Δt ($\Delta t \rightarrow 0$). Obviously, if the total number waiting is less than C (for example total number waiting for the $k+3^{th}$ bus) then all can board the bus that arrives ($k+3^{th}$ bus) and the total number of passengers starts at zero from time = d_i^{k+1} + Δt ($\Delta t \rightarrow 0$). Although not relevant to this discussion it may be noted that not all non-transferring passengers who arrive can board their intended bus (for example, all who arrive before d_i^{k+1} cannot board the $k+1^{th}$ bus); Only those who arrive before t_a (which can be obtained from the bottom figure) can board the $k+1^{th}$ bus. Further transferring passengers

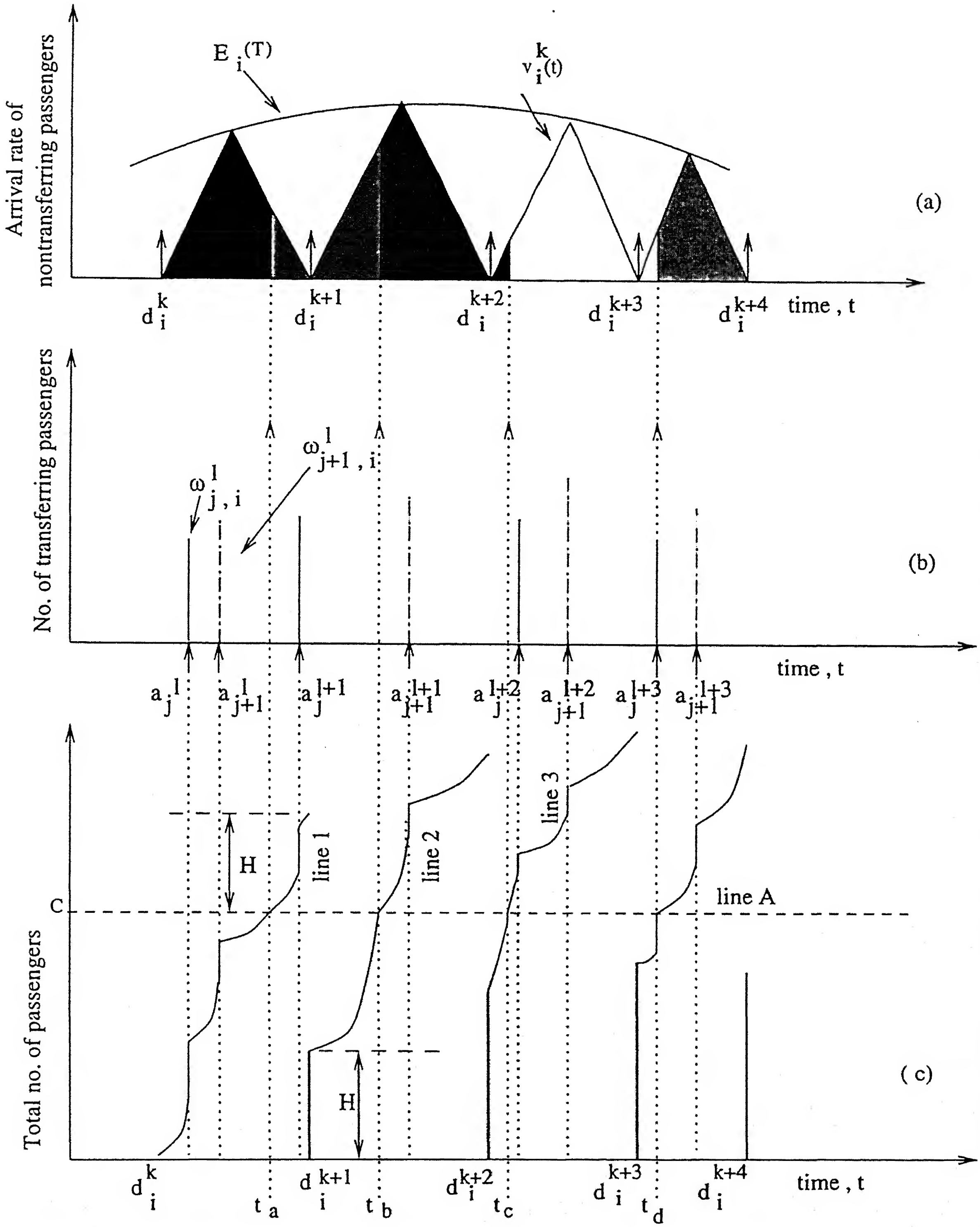

Figure 3.1: Queueing pattern of passengers at station with capacity constraint

cannot always board the next available bus on the route of their choice. For example, those who arrive on the $l+1^{th}$ bus of the j^{th} route and want to transfer to route i, cannot transfer to the $k+1^{th}$ bus of the i^{th} route. This however could have been possible if capacity of bus is large enough, since $l+1^{th}$ bus of the route j arrives before the $k+1^{th}$ bus of route i departs. The above fact creates large difficulties (which are possibly unsurmountable) in trying to obtain an MP formulation of the capacity - constrained problem.

It should be noted that such queuing do take place at transit stations where the demand for a bus often exceeds its capacity. Hence, the above description is a reasonably realistic picture of passenger movement on a transit system.

3.2.1 MP Formulation

In order to develop a mathematical programming formulation for obtaining optimal schedules one represent the above scenario through function in the form of the objective or constraints. First an attempt is made to develop the objective function representing TWT for this problem. This is then followed by a discussion on the constraints.

Objective function:

As before, the objective is to reduce the TWT which is the sum of IWT and TT. In order to understand the difficulties in obtaining a generalized function for the objective consider the example scenario shown in Figure 3.3. For the time window shown in the figure the TT term may be written as

$$TT = (d_i^{k+1} - a_j^l)w_{j,i}^l + (d_i^{k+1} - a_{j+1}^l)w_{j+1,i}^l \qquad (3.5)$$

Hence for the entire scheduling time window the TT is:

$$TT = ... + (d_i^{k+1} - a_j^l)w_{j,i}^l + (d_i^{k+1} - a_{j+1}^l)w_{j+1,i}^l + ... \quad (3.1)$$

Note that transfers are no longer, necessarily to the next available bus (as discussed earlier). Hence the use of δ_{ij}^{kl} variables and the constraints G5 and G6 do not serve the purpose of identifying to which bus (the value of l) of the j^{th} route, the passengers from the k^{th} bus of the i^{th} bus can transfer. This is a major hurdle, because now one cannot obtain a generalized function for TT.

Similarly for the example shown in Figure 3.3, the IWT for all i, k can be written as:

$$\begin{aligned} IWT &= \int_{d_i^k}^{t_a} v_{i,k}(t)(d_i^{k+1} - d_i^k - t)dt + \int_{t_a}^{d_i^{k+1}} v_{i,k}(t)(d_i^{k+2} - t_a - t)dt + \\ & \int_{d_i^{k+1}}^{t_b} v_{i,k+1}(t)(d_i^{k+2} - d_i^{k+1} - t)dt + ... \end{aligned} \quad (3.2)$$

For the entire scheduling time window the IWT term (for this example and for all i, k) can then be written as:

$$\begin{aligned} IWT &= ..... + \int_{d_i^k}^{t_a} v_{i,k}(t)(d_i^{k+1} - d_i^k - t)dt + \int_{t_a}^{d_i^{k+1}} v_{i,k}(t)(d_i^{k+2} - t_a - t)dt + \\ & \int_{d_i^{k+1}}^{t_b} v_{i,k+1}(t)(d_i^{k+2} - d_i^{k+1} - t)dt + ... \end{aligned} \quad (3.3)$$

Again the problem in obtaining a generalized description for the IWT term is the presence of t_a, t_b, etc in the expression of IWT. It can be easily seen

that obtaining general expression to determine t_a, t_b etc are difficult. This is so, because one has to devise a function to enmurate the process described in Figure 3.3 ; note that using this figure the values t_a, t_b etc. can be obtained as the value of the abscissa where the line "number of passengers waiting = C" (line A) intersects the lines 1, 2, etc. Even if one can obtain such a function, using them in the IWT expression will make the objective a highly complicated expression.

The above discussions amply illustrate the fact that an MP formulation for the capacity - constrained problem is almost impossible. Even if one can obtain such a formulation, it will be much complex then that obtained in the previous section. As reported in earlier discussion, traditional techniques failed to obtain an optimal solution even for the earlier MP formulation. One can then safely infer that traditional means would also fail in obtaining an optimal schedule for the capacity - constrainted problem.

It has been reported else where (Chakroborty et al. [1]) that *Genetic algorithms*(GAs) a non-traditional optimization tool performed well to obtain the optimal solution for the problem with the assumption of infinite bus capacity. Hence, in this work, an attempt is made to use GAs to obtain the optimal solution. The next two chapters introduces GAs and develops a formulation of the present problem which can be used by GAs to obtain optimal schedules.

Chapter 4

Genetic Algorithms

4.1 Introduction

Genetic algorithms (GAs) were developed by John Holland and his colleagues at the university of Michigan in 1965. GAs are adaptive search and optimization algorithms based on the mechanics of natural selection and natural genetics (Goldberg et al. [7]).

The discussion related to GA is adapted from the Subrahmanyam [5], Srinivas [3], Deb [8] and Goldberg [7]. Concept of GA is based upon the Darwin's theory of survival-of-the-fittest among string structures. Genetic algorithms make it possible to explore a far greater range of potential solution to a problem than do conventional methods . They do not require any information like continuity, existence of derivatives, and unimodality of an objective function. In every generation (explained later) a new set of artificial creatures(strings) is created using bits and pieces of fittest of the old; an occasional new part is tried for good measure. Genetic algorithms has remarkable ability to focus their attention on the most promising parts of a solution space is a direct outcome of their ability to combine strings containing parts of a partial solutions . The Genetic algorithms exploits the higher pay-off or target

regions of the solutions of the solution space because successive generation of reproduction and crossover produces increasing numbers of strings in those regions. The GA is a search procedure that uses random choice as a tool to guide a highly exploitative search through a coding parameter space.

Genetic algorithms are theoretically and empirically proven to provide robust search in complex spaces. These algorithms are computationally simple yet powerful in their search for improvement. Further, they are not fundamentally limited by restrictive assumptions about the search space (assumptions concerning continuity, existence of derivatives, unimodality and other matters).

4.2 How GA is Different from Traditional Methods ?

Genetic algorithms are different from traditional optimization and search procedures in four ways (Goldberg, 1989).

1. GAs work with a coding of the parameter set, not the parameters themselves.
2. GAs search from a population of points, not from a single point.
3. GAs use payoff (objective function) information, not derivatives or other auxiliary knowledge.
4. GAs use probabilistic transition rules, not deterministic rules.

Many search techniques require auxiliary information to direct their search. For example, Gradient Techniques require derivatives in order to climb the current peak. However GAs do not require any such kind of information. They remain general by exploiting information available in any search direction. To

perform an effective search, they require only payoff values associated with individual strings.

The basic problem with most of the traditional methods is that there are fixed transition rules to move from one point to another. So these methods can only be applied to special class of problems where any point leads to the desired optimum. That is why these methods are not robust and simply can not be applied to a wide variety of problems. But the GAs use probabilistic rules to guide their search i.e. the transition rules of GAs are stochastic. However, a distinction exists between the randomized operators of GAs (explained in the next section) and other methods that are simple random walks. Genetic algorithms use random choice to guide a highly exploitative search towards regions of search space with likely improvement. This characteristic of GAs permits them to be applied to a wide class of problems giving them the robustness that is very useful in very sparse nature of engineering design problems.

In many conventional optimization methods, a single point in the decision space is selected first and some transition rules are used to move to the next point. This point to point method is not always correct because it may locate a false peak i.e. local optimum in multimodal search spaces. But GAs start from a population of points simultaneously climbing many peaks in parallel, thus the probability of finding a false peak is reduced. Because there are many strings that are processed simultaneously and used to update any string in the population, it is very likely that the expected GA solution may be global solution.

Robustness i.e. the balance between the efficiency and efficacy necessary for survival many different environments makes them wide spread applications and resulting advantages over other commonly used techniques. The working principle of GAs are briefly explained below.

4.3 Working principle

Working principle of GAs are very different from that of most of traditional optimization techniques. Genetic algorithms require the natural parameter set of the optimization problem to be coded as a finite length string over some finite alphabet. It begins with a population of strings created randomly. Binary coded strings are used mostly. The length of string is usually determined according to the accuracy of the solution desired. Thereafter each string in the population is evaluated. The population is then operated by three main operators - reproduction, crossover and mutation - to create a hopefully better population. The population is further evaluated and tested for termination. If the termination criteria are not met, the population is again operated by the above three operators and evaluated. This process is repeated until termination criterion is met (Deb, 1995).

```
Begin
    Initialize population;
    Evaluate population;
    repeat
            Reproduction;
            Crossover;
            Mutation;
    until (termination criteria);
End.
```

Figure 4.1 : A pseudo-code for a simple genetic algorithm

4.3.1 Strings

Strings of artificial genetic systems are analogous to chromosomes in biological systems. In natural chromosomes, we have genes which influence the physical characteristics of an individual. Similarly, in artificial genetic systems, chromosomes are composed of genes which take some number of values called alleles. The values of 1 or 0 in each position of a binary string influence the coordinates of a point in search space as each string represents all the problem variables in the problem under consideration.

For application of GA techniques, an initial population of strings is required. Out of the number of ways available for generation of this initial population of strings, technique of tossing of an unbiased coin is quite popular. In this technique, successive coin flips (head = 1, tail =0) are used to decide the genes in the string. This procedure is continued till the strings are generated in the initial random population. The length for a string i.e. the total number of bits (genes) is equal to the sum of the bits assumed for each of the problem variables in the optimization problem. The number of bits for a variable depends on the on the precision required.

4.3.2 Coding and Decoding

We know that while using traditional optimization techniques we use the variables themselves. It may be appreciated that while working with GAs, we work with coding of variables (instead of the variables themselves). A successful method of coding multiparameter optimization problems of real parameters is the concatenated, multi parameter, mapped fixed-point coding. In multivariable optimization, all the problem variables are coded in a single string and each variable has a specific location in the string. Most commonly used coding in GA applications is the binary coding. The precision of problem variables

depends on the number of bits assumed for them in the string and the actual range in which they vary.

A string represents a point in the space and the decoded values of the string's contents will represent the coordinates. For example, if an integer variable varies from 20 to 51, it can be completely represented by a five digit string which can give 32 unique strings. The decoded value for string 00000 is 0, where as the decoded value for 11111 is 31, remaining all strings will have decoded values in between 0 and 31. Consider a string 10100. The decoded value for this string is $(0 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 0 \times 2^3 + 1 \times 2^4)$ i.e. 20.

4.3.3 Evaluation

The decoded values from a string varies from 0 to 2^l where l is the number of bits in the string. However, the actual values may not vary in that range. Then, the decoded values obtained from the strings should be mapped in the range in which the actual values will vary. After decoding all the variables, they should be mapped as given below. Let lb and ub be the actual lower and upper bounds in which a variable varies and dc be the decoded value of the string, then its mapped value is given by the following equation.

$$Mapped\ value = lb + \frac{(ub - lb) \times dc}{2^l - 1} \tag{4.1}$$

The precision of the variable is $\frac{(ub-lb)}{2^l-1}$

After mapping of all the variables, they can be used to calculate the objective function value (also called fitness value). Accordingly, fitness values are calculated for all the strings in a particular generation. Then the termination criteria is checked. In case the termination criteria is not reached, the GA operators are applied on the present population to get a new set of population.

4.3.4 GA operators

A simple genetic algorithm that yields good results in many practical problems is composed of three GA operators: Reproduction, Crossover and Mutation. These operators are applied to successive string populations to create new string populations. Application of these operators is very simple and involves nothing more than random number generation, string copying and partial string exchanging. It may be noted that despite their simplicity, the resulting search performance is wide ranging and impressive. Each of these three operators is described below.

4.3.4.1 Reproduction

It is the first operator applied on a population. In this process individual strings are copied according to their objective function values, which means that the strings with a better value will have a higher probability of contributing one or more strings in the next generation. The operator is an artificial version of natural selection, a Darwinian survival of the fittest among string structures. The best string in the present generation will get more copies, the average stay even and the worst will die off in the next generation. It selects good strings in the population and forms a mating pool. There exists many reproduction operators in GA literature, but the essential idea of each is that above-average strings are picked from the population and their duplicates are inserted in the mating pool. There are several selection schemes available for use these day, and a detailed comparative analysis is made in (Goldebrg, [7]). The Tournament Selection operator has been used and the same is described in the following paragraph.

In Tournament Selection, some strings are picked at random and the better string is copied to the mating pool. This process is repeated till the

required number of individuals equal to the population size are selected. The number of individuals considered for the selection of an individual is known as *Tournament size.* A tournament size of 2 is generally used in many applications and the same has been used in our study. In this selection, two individuals are chosen at random from the current population and the better amongest these is selected with fixed probability ranging from 0.5 to 1.0. Tournament selection can be used for both minimization and maximization problems without any transformation of the objective function. The strings thus selected by reproduction are kept in mating pool for applying the next operator, Crossover.

4.3.4.2 Crossover

After reproduction, now the population is enriched with good strings from the previous generation, but it does not have any new string. The basic idea of applying Crossover operator to this new population is to create better strings. Some of the better strings are preserved as found previously. Crossover is usually performed on only some of the strings in the mating pool. The total number of strings to participate in crossover is controlled by the *Crossover probability,* which is the ratio of total number of strings selected for mating and the population size. The crossover probability is usually kept high, which varies from 0.90 to 0.95 in many applications.

There are a number of crossover operators in use today, like single point crossover, multi point crossover, and uniform crossover. In all these operators, two strings are picked from the mating pool at random and some portion of strings are exchanged between the strings. In the single point crossover, this is performed by choosing a random site along the string and by exchanging all bits to the right of the crossing site as shown below.

$$\begin{array}{ccc} 111|01 & & 111|10 \\ & \Longrightarrow & \\ 110|10 & & 110|01 \end{array}$$

The crossover operator searches the parametric space by exchanging the information between two strings. The strings selected for applying crossover operator are called the parent strings and the strings obtained after crossover are known as child strings.

If appropriate site is selected, good child strings are obtained. However, as this site is not known beforehand, a random site is selected. With random site, the child strings may or may not be good. If the string is good, it is selected. The bad strings created may not survive beyond next generation due to the application of reproduction operator in the next generation. The off spring do not replace the parent strings instead they replace low fitness strings which are discarded at each generation so that the total population size remain the same. Crossover operator is mainly responsible for the search aspect of genetic algorithm.

4.3.4.3 Mutation

In GA operators, Mutation plays a secondary role. Although reproduction and crossover efficiently exploit the search space but, sometimes they lose some potentially useful genetic material (1's or 0's at particular locations). To compensate this loss, mutation operator is used which changes a 1 to 0 and vice versa with a small *Mutation probability.* The need for mutation is to keep diversity in the population. It is done on bit-by-bit basis. Since this operator disrupts a string, the mutation probability is kept very low.

To further understand the role of Mutation operators, let us consider an example.

0111

0011

0101

1101

Let the optimal string be 1110. By reproduction and crossover we will never be able to get the optimal string. But mutation may bring 0 in the last position of 1111 resulting in 1110. Then finally the problem will converge to the string 1110.

After applying all the above GA operators, a new set of population is created. Then, they are decoded and objective function values are calculated. This completes one generation of GA. Such iterations are continued till the termination criteria are reached.

4.3.5 Termination Criteria

When the average fitness of all the strings in a population is nearly equal to the best fitness, the population is said to have converged. When the population is converged, the GA is terminated. The same can be done by fixing maximum number of generations, the number of generations at which population will converge. In genetic algorithms, maximum number of generations is generally used as the termination criteria. The same is used in the present study.

Chapter 5

Revised Formulation

The traditional formulation of the scheduling problem was introduced in Chapter 3, while in this chapter, the revised formulation of the problem is presented. This formulation assumes that GAs (described in Chapter 4) will be used as a solution technique. The first section of the chapter deals with the revised formulation of the problem, while the second deals with the implementation details.

5.1 Revised Formulation

As stated in Chapter 3, obtaining a traditional formulation for the finite bus capacity version of the scheduling problem is almost impossible. The difficulties arise in trying to represent through a functions, elaborate procedures of obtaining which passenger boards which bus. Hence, in the reformulation, procedure based declarations (as offered to function based descriptions) are used to determine which passenger will board which bus and consequently the IWT and TT for a given schedule. The inclusion of procedure based declarations in the formulation is possible since GAs can work with such declarations while obtaining the optimal solution. Further, features like coding of variables also

help us in reducing the complexity of the problem.

Before proceeding further, it is beneficial to take a look at the infinite - bus - capacity formulation (see equation 3.1). This is because some of the constraints mentioned there are relevant to the finite bus capacity problem also. These constraints fall under the following two classes:

1. Type I : Variable bounds (Constraints G1, G2, and G3) and
2. Type II : Maximum transfer time (Constraint G4)

Constraints G1, G2 provide the upper and lower bounds on stopping time respectively, while constraint G3 provides an upper bound on the headway between two consecutive buses. These constraints obviously remain even for the finite bus capacity problem. However, in the revised formulation these two sets of constraints are effectively eliminated by cleverly choosing decision variables. In this revised formulation the decision variables are no longer arrival times and departure times of buses. Stopping times dx_i^k (dx_i^k = d_i^k - a_i^k; the stopping time of k^{th} bus of the i^{th} route) and the headways x_i^k(x_i^k = a_i^k - a_i^{k-1} ;the headway between k^{th} and $(k-1)^{th}$ bus of the i^{th} route) are used as the decision variables.

In GAs the decision variables are coded as fixed line binary strings. These binary strings are assumed to map into a given range for the variables. Hence, by coding x_i^k and dx_i^k and mapping then into the range 0 to h_i, and s_i^{min} to s_i^{max}, respectively one effectively eliminate constraints G1, G2 and G3.

Constraints G4 represent the service related requirement, that transfer time can not be more than certain value. However, in the finite bus capacity problem the form of these constraints (as shown in equation 3.1) have to be different so the variable δ_{ij}^{kl} do not serve the intended purpose any more. As mentioned earlier an equivalent functional form for this constraints in the

finite bus capacity case is difficult. Hence, in the revised formulation these constraints are taken care of through an external procedure, which computes all transfer time for a given schedule and added penalty whenever the transfer time is more than specified upper bound. It may be noted here that constraints G5 and G6 do not serve their purpose for the finite bus capacity problem.

The objective function, in the revised formulation is computed using a procedure which emulates the process described in section 3.2. In summary the following may be noted:

1. The decision variables are, stopping times (dx_i^k) and headways (x_i^k).

2. The service related constraints of minimum stopping time, maximum stopping and policy headways are handled using the coding and mapping of variables allowed in GAs.

3. Individual transfer times and initial waiting times are calculated from external procedure which emulates the process as described in section 3.2.

   (a) The service related constraints of maximum transfer time are taken into account by adding a penalty to the objective value whenever are individual transfer time is greater than maximum allowable value.

   (b) The total initial waiting time and the total transfer time (sum of which form the objective) are obtain by summing the individual initial waiting time and transfer time, respectively.

The revised formulation becomes clearer after going through implementational details described in the next section.

5.2 Implementation

The implementation of the above reformulated problem algorithms is presented in this section. In order to understand the implementation of the scheduling problem a flow diagram of the process is shown in the Figure 5.1. Each task in the figure is classified into one of six classes (Class I to VI). The classification of the task is written in roman numerals in the figure. In the following each of these six classes are discussed in details.

Class I

In Class I an initial population of strings are created randomly. A string is obtained as a conglomeration of substrings which represent all the headways and stopping times of all the routes. Such a string, therefore, represents the entire schedule.

If the total number of buses, plying on i^{th} route is n_i, then there will be $n_i - 1$ headways (because $\Sigma_i n_i$= Scheduling time window) and n_i stopping times. Here, the stopping time of all the buses at a particular route are assumed to be equal, i.e., $dx_i^k = dx_i$, Hence, there is only one variable for the stopping time for a route at a station. Therefore, the total number of variables for each route is equal to n_i-1+1 $= n_i$.

The problem variables headways and stopping times x_i^k and dx_i^k respectively, take only discrete values over a specified range. Although, by selecting the appropriate number of bits representing the variable one can achieve the desired precision for that variable. For example if a variable, x_i^k lies over the range ρ_i^k then the desired precision is p the number of bits(α_i^k) required to code the variable is (i.e., the substring length representing the variable)

$$\alpha_i^k = \log_2(\rho_i^k/p) \tag{5.1}$$

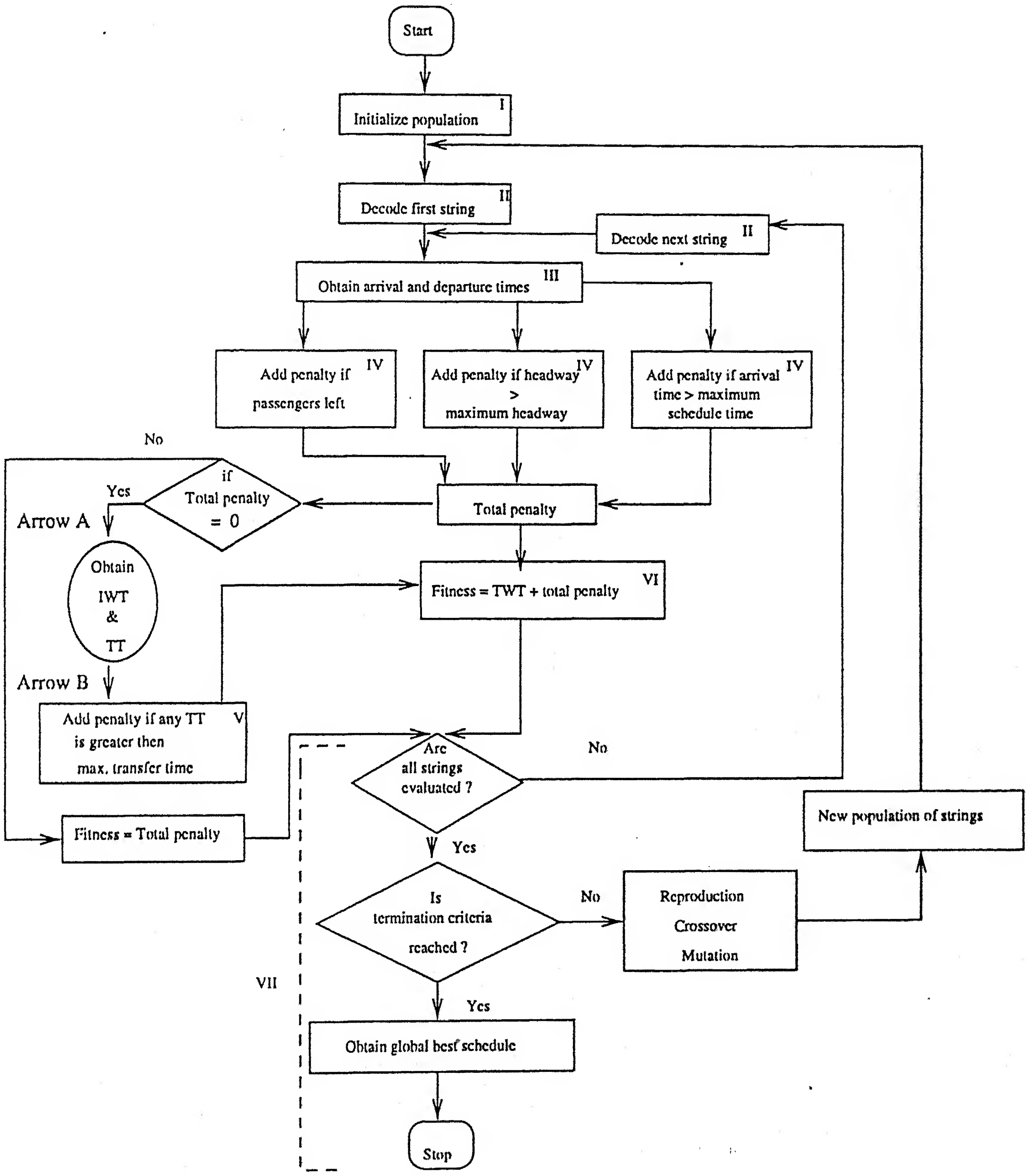

Figure 5.1: Implementation of the scheduling problem using GAs

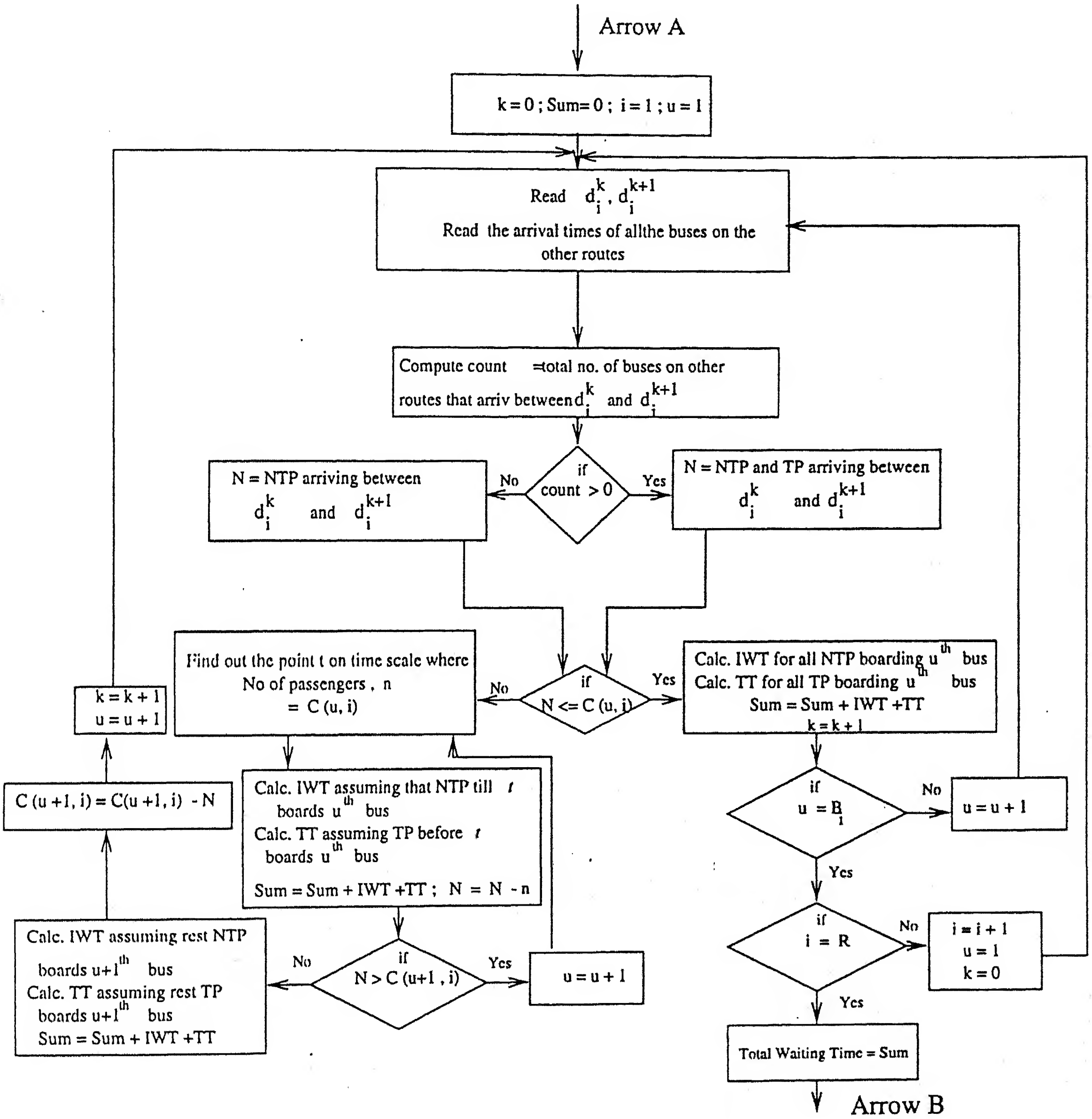

Figure 5.2: Calculation of TWT

The process of obtaining the substring length for a variable is further illustrated through the following example on determining the length for a headway, n_i^k. If ρ_i^k is 32 minutes (say, from 5 to 36) and p is chosen to be 0.5 minutes then α_i^k, the substring length required, is $\log_2 32/0.5$ or 6.

Thus the total string length required for a complete schedule is $I.\Sigma_{k=1}^{n_i-1}\alpha_i^k+\beta_i$; where I is the total number of routes and β_i is the number of bites required to represent the stopping time.

Class II

In Class II task, for each string all its substrings are decoded and the corresponding real values are obtained. These are then mapped into the variable range to determine the actual parameter value. For example, consider a two bit substring 11, representing the stopping time. Then the decoded value of the substring is 3 $(1 \times 2^1 + 1 \times 2^0)$ and the actual parameter values is $3 + 2 = 5$, where 2 is the lower limit of the stopping time.

Class III

This task calculates the scheduled arrival times (a_i^k) and the scheduled departure times (d_i^k) for all the buses on all the routes from the values of headways and the stopping times using the procedure shown below.

Begin

for all routes (i=1 to r)

$a_i^0 = 0$

for all buses (k=*1 to* $n_i - 1$) *compute*

$a_i^k = a_i^{k-1} + x_i^k$

$d_i^k = a_i^k + dx_i^k$

End.

Note that since the schedule time window is fixed to P the last bus must depart at S + P. Where S is the starting time of the scheduling time window.

Class IV

In the tasks of this class, certain undesirable/infeasible schedules are identified and penalized.

(a) It is decided that the schedule should be such that at the end of the scheduling period no passengers are left unattended. Hence if a schedule is such that the total number

Initialize

Begin

for all i

Begin

$d_i^0 = 0$

for all k

$MC_i^k = C_i^k$

End

for all routes (i=1 to r)

Begin

for all buses (k=1 to n_i)

total transf. pass. $= 0$

for all $I \neq i$

Begin

for all l

if $(d_i^{k-1} \leq a_I^l \leq d_i^k)$

total transf. pass. $=$ *total transf. pass.* $+ \omega_{Ii}^l$

compute total non-transf. pass. (refer chapter 3 for detail)

total pass. $=$ *total transf. pass.* $+$ *total non-transf. pass.*

End

if $MC_i^k \geq$ *total pass. then*

left = *0*

else

if $k = n_i$ *then obtain penalty L*

else

$MC_i^{k+1} = MC_i^{k+1}$ - (*total pass.* - MC_i^k)

End

End.

of passengers waiting for last bus cannot be accommodated in the last bus then that schedule is considered undesirable and a penalty value is computed for it. Such schedules are identified and penalized using a procedure based on the logic described above. The notation used above are:

C_i^k :	Capacity of k^{th} bus on i^{th} route.
MC_i^k :	Modified Capacity of k^{th} bus on i^{th} route.
r :	Total number of routes.

(b) The last headway, for each route is computed by subtracting the sum of all other headways on that route from the scheduling time window (because departure time of last bus is fixed). Thus in certain schedules the last headway may be more than the policy headway. If so a penalty (called penalty H) is computed.

(c) Sometimes it may happen that the sum of all the headways (other than the last headway) exceeds the scheduling time window; i.e. departure time for some bus may be greater then S + P. This is again an infeasible schedule. For such schedules a penalty (called penalty D) is computed.

Class V

In the task of this class the total transfer time, TT and the total initial waiting time, IWT for each of the schedules (note every string in the population represents a schedule) in the population is calculated. As explained earlier computing TT and IWT is quite complex in the finite bus capacity version of scheduling problem. In the present formulation an external procedure based declaration therefore is used to compute IWT and TT. This procedure is presented as a flow chart in Figure 5.2. It may be noted that the flow chart basically implements the computation process explained in section(3.2) of chapter 3. The notation used in the figure are:

B_i : Total number of buses on route i.

C(u, i) : Capacity of u^{th} bus on i^{th} route.

R : Total number of routes.

n : Total number of passengers (both transferring and non-transferring) who arrive between any two consecutive buses of a particular route.

NTP stands for non-transferring passengers, TP stands for transferring passengers, k, u indicates the bus number for any route and i indicates the route number.

In this class of activities a penalty term is also compute whenever, the transfer time for a individual exceeds the maximum allowable value. This penalty term is called T.

Class VI

In this class of activity the fitness for each string representing a schedule is obtained. The process of obtaining is evident from figure 5.2.

Class VII

This class of tasks encompass all the GA related activities. After obtaining the fitnesses for all the strings the termination condition is checked i.e.

before going into creating new population of strings using GA operators. It is first checked whether the population has converged to a solution. If it has then the best string from this population is selected and reported as a best schedule. This ends the optimization process. However if the population is not deemed to have converged then the three GA operators, Reproduction, Crossover, and Mutation are used in sequence to obtain a new and better population of strings (schedules). With this new population the processes from Class II tasks start all over again.

Chapter 6

Results

In this chapter, optimal schedules for number of test cases using the GA based procedure are presented. Before presenting the result, however, the GA parameters and the problem parameters are presented. It may be noted that although the following parameters are used, the procedure developed earlier is general and can work for or with a different set of parameter values.

6.1 GA Parameters and Problem Parameters

6.1.1 GA Parameters

- Population size = 600 or 700
- Maximum number of generations = 1200
- Cross over probability = 0.95; single point crossover is used
- Mutation Probability = 0.005; bitwise mutation is used
- Binary tournament selection is used
- Number of bits used for coding x_i^k (headways) = 5

- Number of bits used for coding dx_i^k (stopping times) = 2
- Binary tournament selection is used
- Total string length = 141

6.1.2 Problem Parameters

- Maximum transfer time = 30 minutes
- Maximum stopping times = 5 minutes
- Minimum stopping times = 2 minutes
- Range of headway :
  1. For first bus on each route = 0 - 31 minutes
  2. For all other buses on each route = 14 - 45 minutes
- Arrival pattern of non-transferring passengers
  1. It is assumed that non-transferring passengers follow some kind of distribution $v_{i,k}(t)$. The function $E_i(\tau)$ represent the functional form of this distribution.

$$E_i(\tau) = (\frac{\tau}{1-\mu})^{1-\mu}(\frac{s-\tau}{\mu})^{\mu} \times \frac{L_i}{s} \tag{6.1}$$

     μ value (referred as MU in the figures) in Equation 6.1 is taken either as 0.2 or 0.5 or 0.8 ; different combinations have been used for different test cases. Also, in some cases $E_i(\tau)$ is assumed to be constant and parallel to the time axis.
  2. The value of L_i in Equation 6.1 is taken as 3.7 in all test cases.
  3. Triangular arrival pattern is used ; that is, $v_i^k(t)$ is triangular for all i and k. The maximum of $v_i^k(t)$ occurs at $t = (d_i^k - d_i^{k-1}) \times 0.75 + d_i^{k-1}$.

- Arrival pattern of transferring passengers

  $\omega_{i,j}^k$ is also assumed to follow some kind of distribution. The functional form of this distribution is:

$$W_{i,j}(\tau) = (\frac{\tau}{1-\eta})^{1-\eta}(\frac{s-\tau}{\eta})^{\eta} \times \frac{\lambda_{i,j}}{s} \tag{6.2}$$

  That is, if the arrival time of k^{th} bus of i^{th} route is a_i^k then number of people waiting to transfer from this bus to j^{th} route is obtained by substituting $\tau = a_i^k$ in the equation $W_{i,j}^k(\tau)$. Different values for η are used in the test cases. The value used are $\eta = 0.2$ for $W_{1,2}(\tau)$, $\eta = 0.3$ for $W_{1,3}(\tau)$, $\eta = 0.4$ for $W_{2,1}(\tau)$, $\eta = 0.3$ for $W_{2,3}(\tau)$, $\eta = 0.5$ for $W_{3,1}(\tau)$ and $\eta = 0.2$ for $W_{3,2}(\tau)$. For a given case all $\lambda_{i,j}$'s are assumed to be same. With respect to transferring passengers the test cases can be grouped into four classes:

  (i) No transferring passengers,

  (ii) Total number of transferring passengers approximately equal to 100,

  (iii) Total number of transferring passengers approximately equal to 200 and,

  (iv) Total number of transferring passengers equal to 120.

  The $\lambda_{i,j}$ value used for class (i) is zero, for class (ii) is 3.0, for class (iii) is 5.2 and for class (iv) $W_{i,j}^k(\tau)$ is assumed to be constant and parallel to the time axis. The value of constant is assumed to be 2.75.

- Maximum schedule time window is s = 240 minutes

- Number of routes = 3

- Number of buses on each route is either 8 or 10 or 12. With respect to the distribution of buses the test cases can be classified into two groups : Group (A) – Equal resources (buses) on each route (number of buses on each route is 10), Group (B) – Unequal resources on each route (number of buses on route 1 is 8, on route 2 is 10 and on route 3 is 12).

- For different test cases different bus capacities are used. The bus capacities used here are 35, 40, 50, 60 and 80. It may be noted that the term "bus capacity" as used here is to be understood as the bus capacity available to the passengers waiting at the station. For example, if a bus with total capacity of hundred arrives with 80 passengers out of which 20 alight then "bus capacity" = 100 - (80 - 20) = 40.

6.2 Optimal Schedules

In this section the optimal schedule obtained using the proposed algorithm are presented. Schedules for a total of 33 test scenarios are presented here. These test scenarios have been classified into 9 cases. For Case I through V, equal resource on each route has been assumed; i.e., number of buses on each route is 10. It may be noted these test cases help to prove the efficacy of the proposed algorithm in obtaining the optimal schedule.

6.2.1 Case I

In this case bus capacity is equal to 40 and arrival pattern of transferring as well as non-transferring passengers is uniform (i.e., $E_i(\tau) = 2.75$ for all i and $W_{i,j}(\tau) = 2.75$ for all $i, j, i \neq j$). For the above parameter values the total number of non-transferring passengers are equal to 980 and total number of transferring passengers is equal to 120 and the total bus capacity available is equal to $30 \times 40 = 1200$. A schedule which is optimal for this case should have the following properties;

(i) All the passengers who arrive during the scheduling time window should be able to leave during this period because total number of passengers arriving is 1100 where as the total bus capacity is 1200.

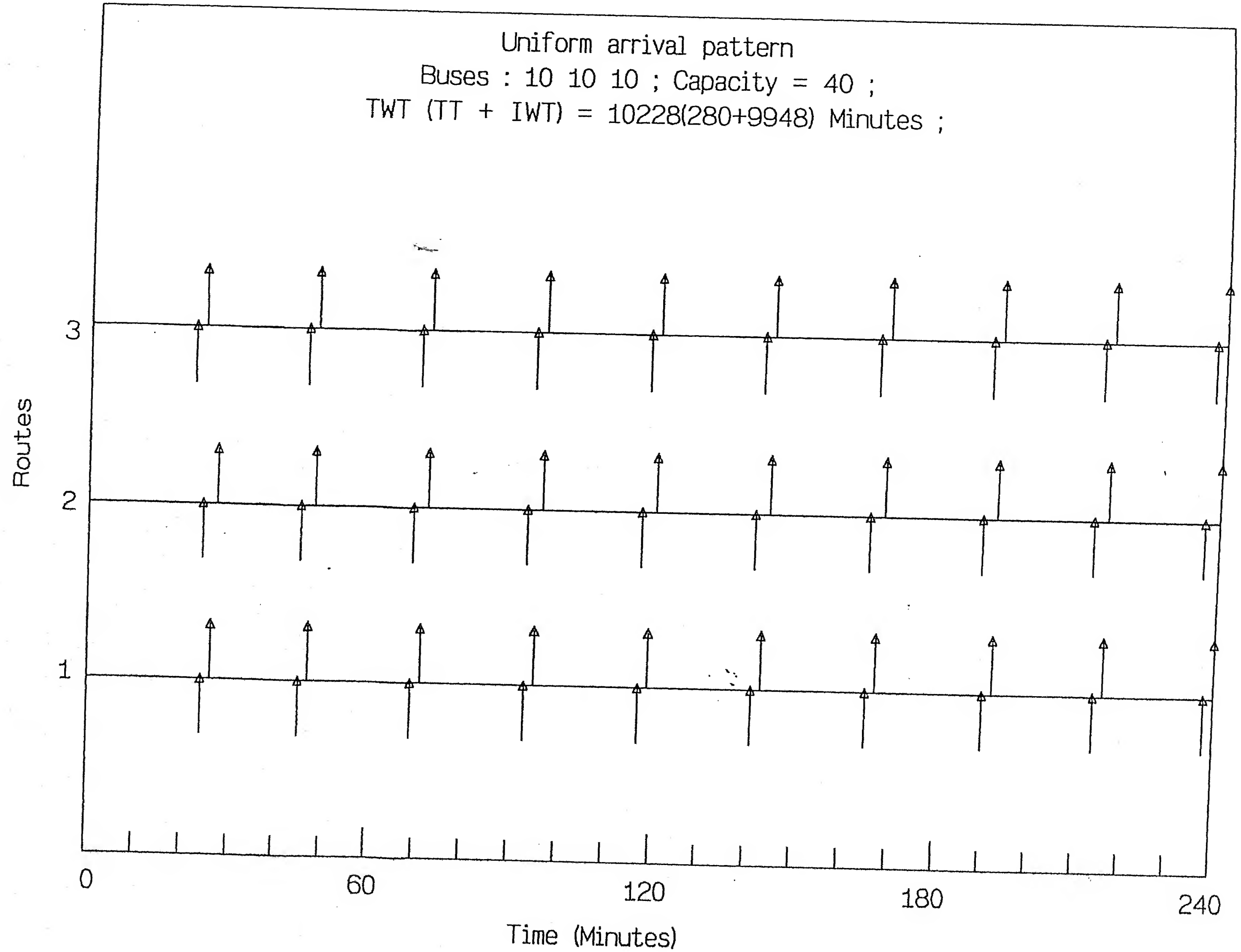

Figure 6.1: Optimal schedule for uniform arrival pattern of passengers and capacity is 40

(ii) The buses on each route should be aligned so that transfer time of passengers is minimal. This property follows from the fact that there are equal number of buses on each route.

(iii) The headways on each route should be equal so that total initial waiting time is minimal. This property follows from the fact that total initial waiting time is proportional to sum of square of headways and the sum of headway is equal to the constant in the present problem.

The optimal schedule is presented in Figure 6.1. As expected all the passengers are accommodated by this schedule, the buses on different routes are aligned and the headways are more or less equal. It may be noted that arrows towards horizontal line represents arrivals and arrows away from the horizontal line represents departures of the buses.

6.2.2 Case II

In this case the value of μ in $E_i(\tau)$, associated with the arrival pattern of non-transferring passengers is 0.2 for all the routes. This means that the arrival rate of passengers in the latter half of the scheduling time period is greater than that in the former half. Further it is assumed there are no transferring passengers.

Under the above condition it is expected that in an optimal schedule the concentration of buses in the latter half will be higher than in the former half. Further the schedule for each route should be same since the demand pattern for each route are same. Figure 6.2 through 6.6 shows the optimal schedules obtained for this case under different assumption of bus capacity. Figure 6.2 is for bus capacity of 35, Figure 6.3 is for bus capacity of 40, Figure 6.4 is for

bus capacity of 50, Figure 6.5 is for bus capacity of 60 and Figure 6.6 is for bus capacity of 80. In all these schedules it can be seen that there is larger concentration of buses during the time period 120 minutes to 240 minutes than during the time period 0 to 120 minutes. Also the schedules for each route are more or less same. Hence the schedules obtained using the proposed procedure does give optimal/near optimal schedules.

The total number of passengers arriving in this case is approximately 970. Obviously then, one would expect a more pronounced effect of the arrival pattern on the schedule when the bus capacity is 35 (i.e., total available capacity $35 \times 30 = 1050$) than the bus capacity is 80 (i.e., total available capacity $= 80 \times 30 = 2400$). Further when the bus capacity is large, one would expect that initial waiting time per passenger would be lesser then when bus capacity small. This is because when the bus capacity is large, there is a greater flexibility in obtaining a feasible schedule (one in which no passenger is left behind) than when bus capacity is small; hence a greater chance of obtaining a schedule which provides lesser waiting time to passengers.

On close examination of schedules presented here one sees that the concentration of buses in the latter half is higher when the bus capacity is equal to 35 than when bus capacity is equal to 80. As expected the initial waiting time per passenger also reduce with increase in bus capacity. This can be seen from Figure 6.7, which shows a plot IWT/passenger versus bus capacity. It is interesting to note that there is a sharp increase in IWT/passenger when bus capacity reduced to 40 to 35. Beyond 40 however there is not much distinction in IWT/passenger. This behavior is also expected because optimal schedule should not be sensitive to how large the excess capacity is.

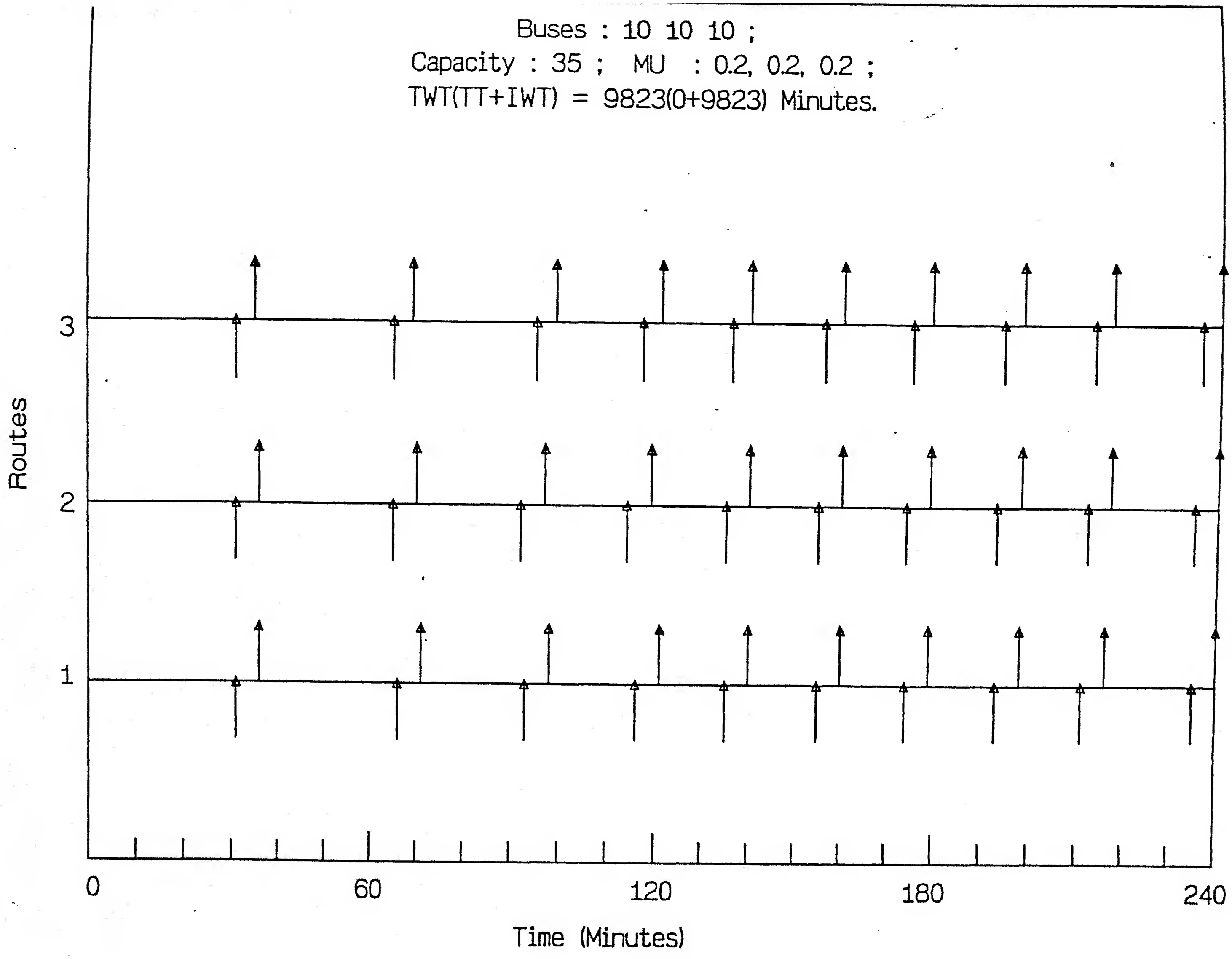

Figure 6.2: Optimal schedule for only IWT consideration with bus capacity 35 (Non-Uniform arrival pattern)

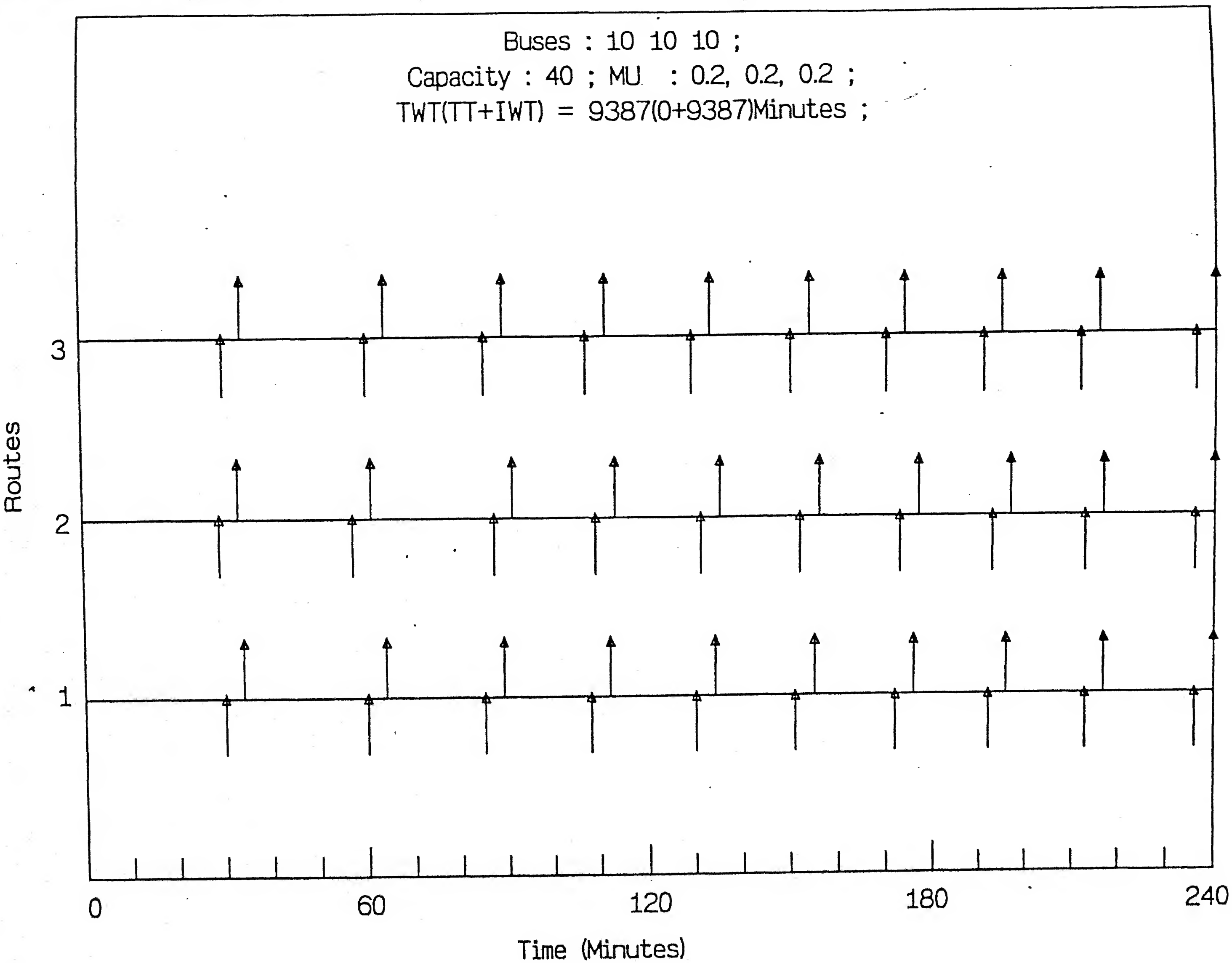

Figure 6.3: Optimal schedule for only IWT consideration with bus capacity 40 (Non-Uniform arrival pattern)

CENTRAL LIBRARY
I. I. T., KANPUR

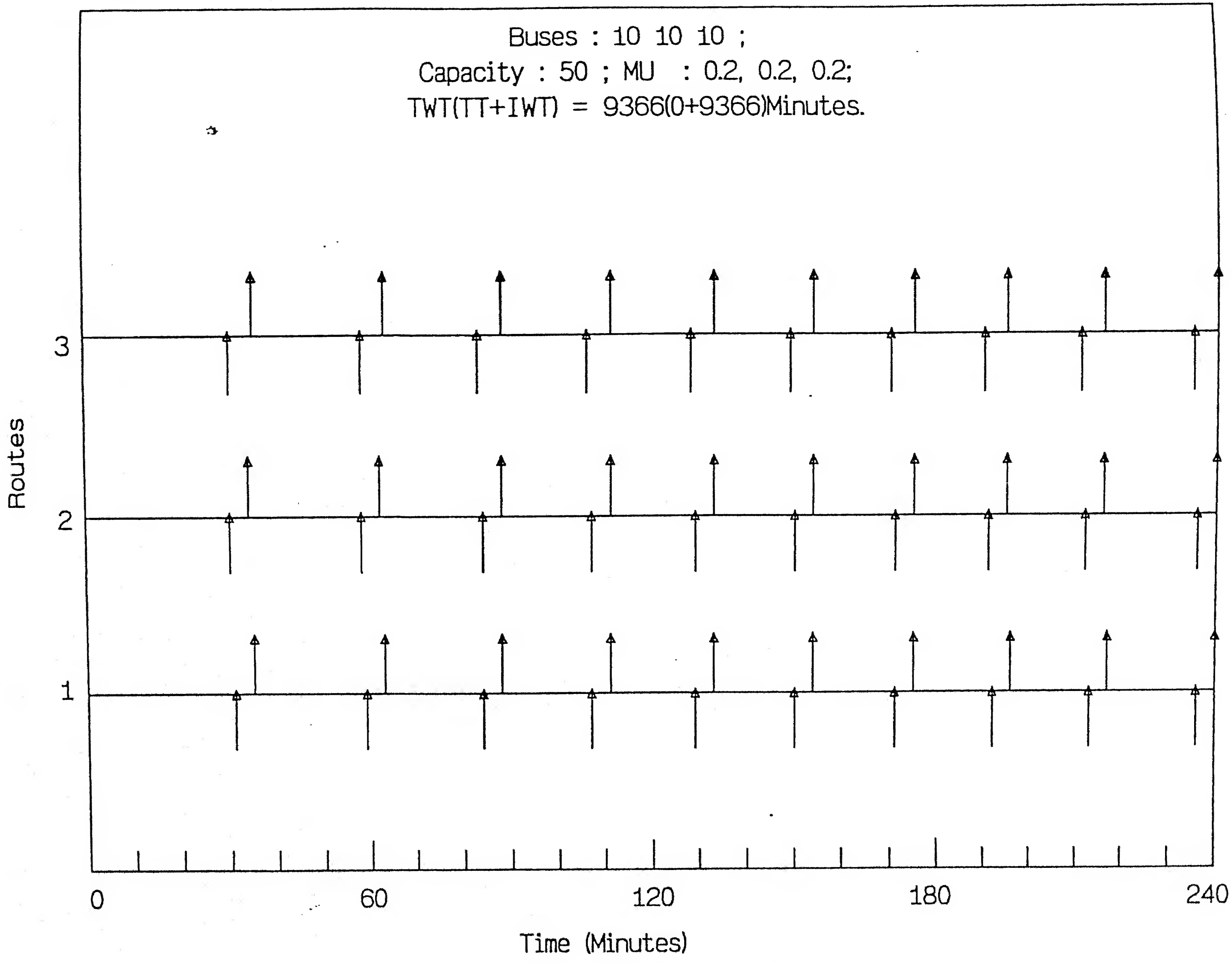

Figure 6.4: Optimal schedule for only IWT consideration with bus capacity 50 (Non-Uniform arrival pattern)

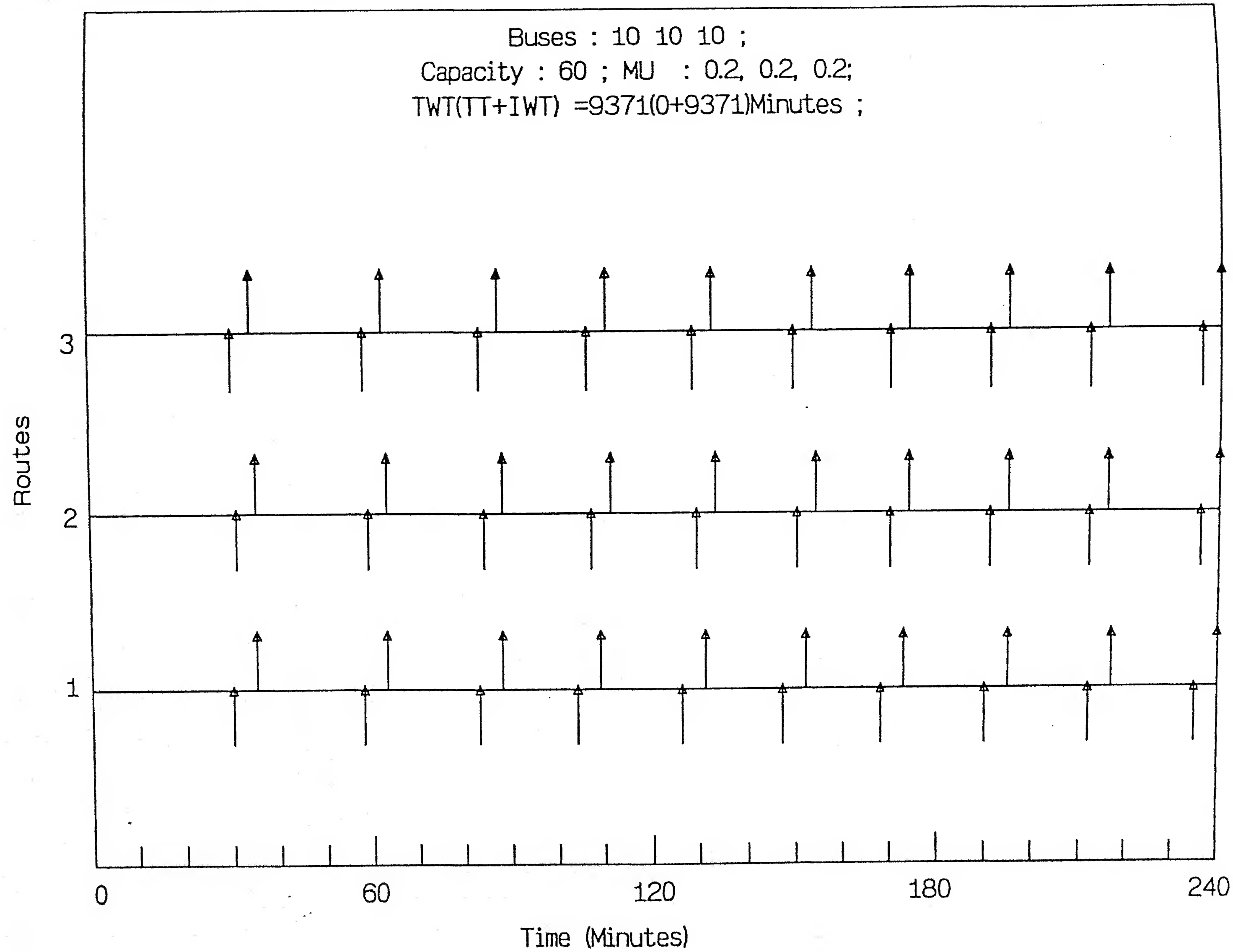

Figure 6.5: Optimal schedule for only IWT consideration with bus capacity 60 (Non-Uniform arrival pattern)

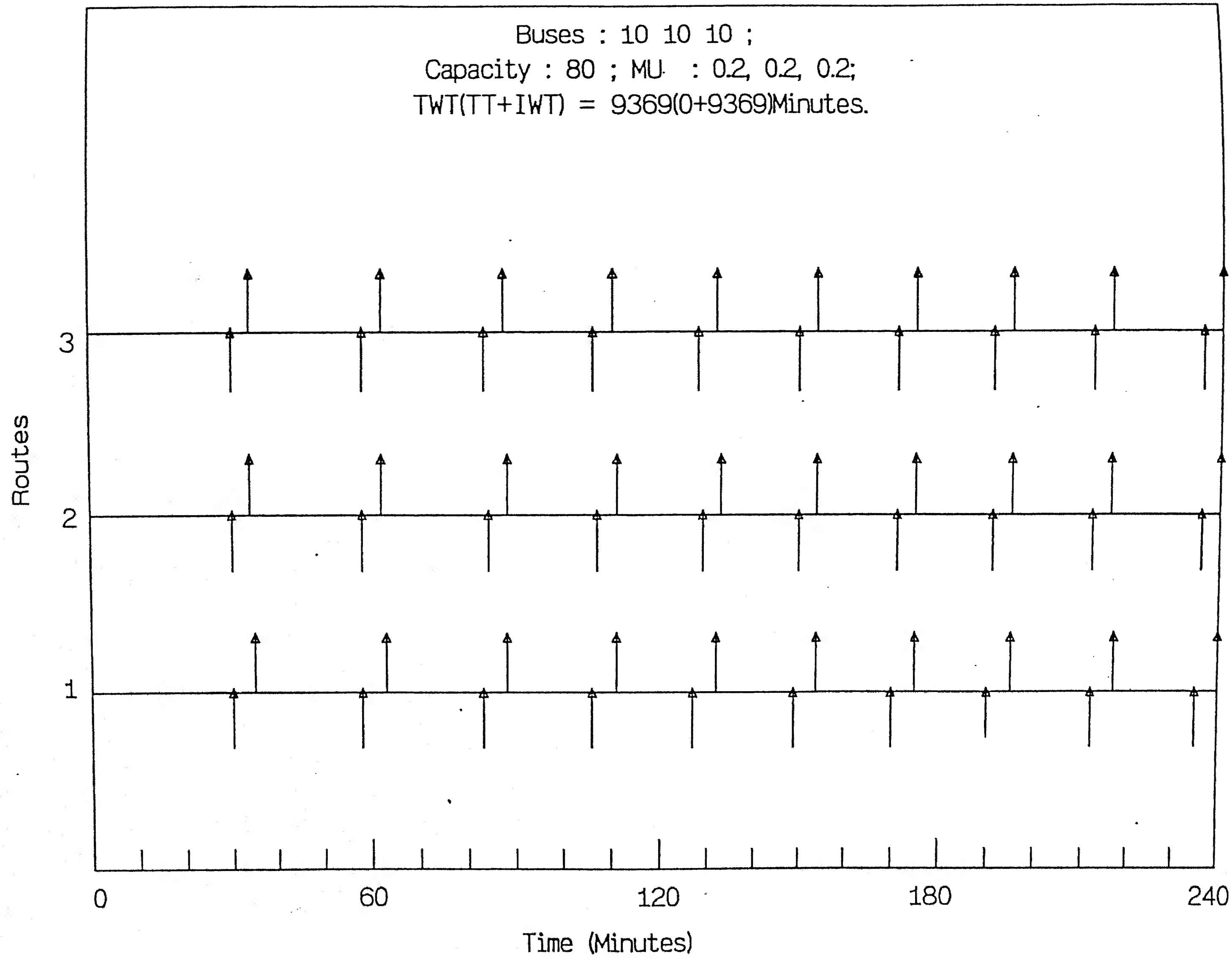

Figure 6.6: Optimal schedule for only IWT consideration with bus capacity 80 (Non-Uniform arrival pattern)

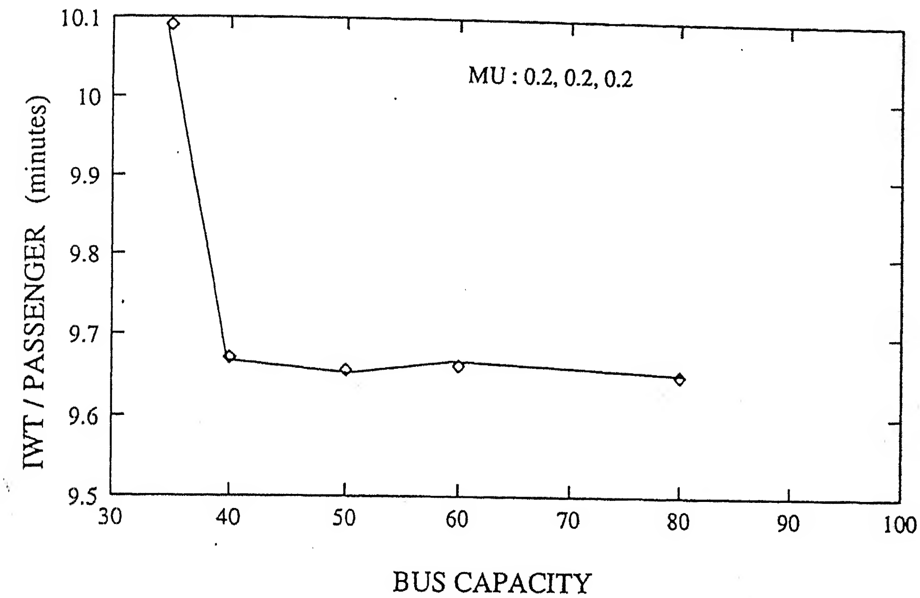

Figure 6.7: Graph for Bus Capacity versus IWT/passengers

6.2.3 Case III

This case is similar to Case II, the only distinction being in the value of μ. A μ value of 0.5 is used for all the routes. This means that the number of passengers coming in the middle part of the scheduling period is greater than the number coming during the other parts of the time period. Hence like earlier one expects a greater concentration of the buses in the middle than at ends. Further, one expects the effect of arrival pattern to be less pronounced in schedules with higher bus capacity.

Figure 6.8 through 6.12 presents the optimal schedules obtained under bus capacity assumptions of 35 to 80. As expected there is a higher concentration of buses in the middle in all the schedules. However, the concentration is slightly higher when bus capacity is 35 than when bus capacity is 80. For example, the time range of middle six buses in Figure 6.8 is 95 minutes while in Figure 6.12 it is 108 minutes.

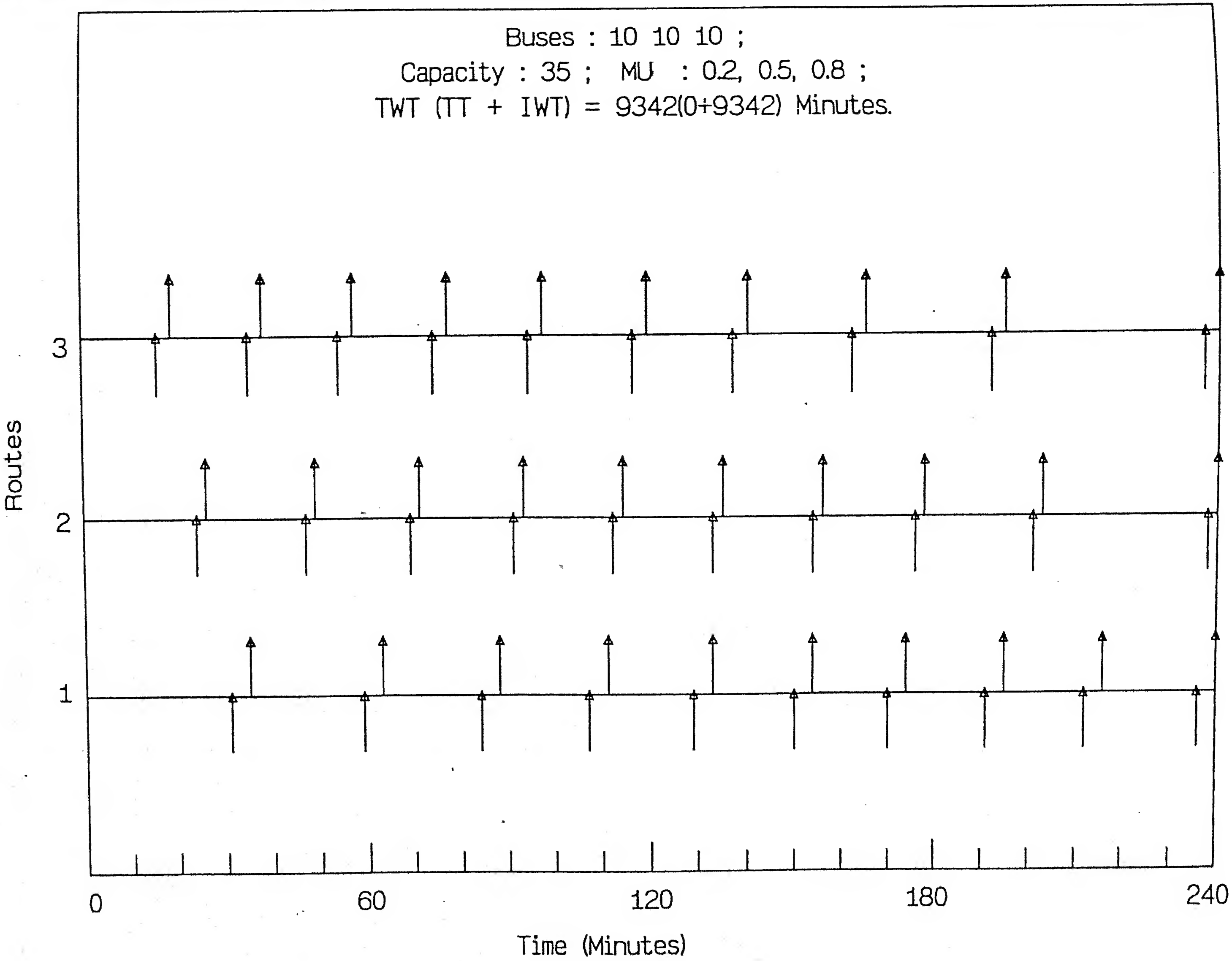

Figure 6.8: Optimal schedule for only IWT consideration with bus capacity 35 (non-uniform arrival pattern)

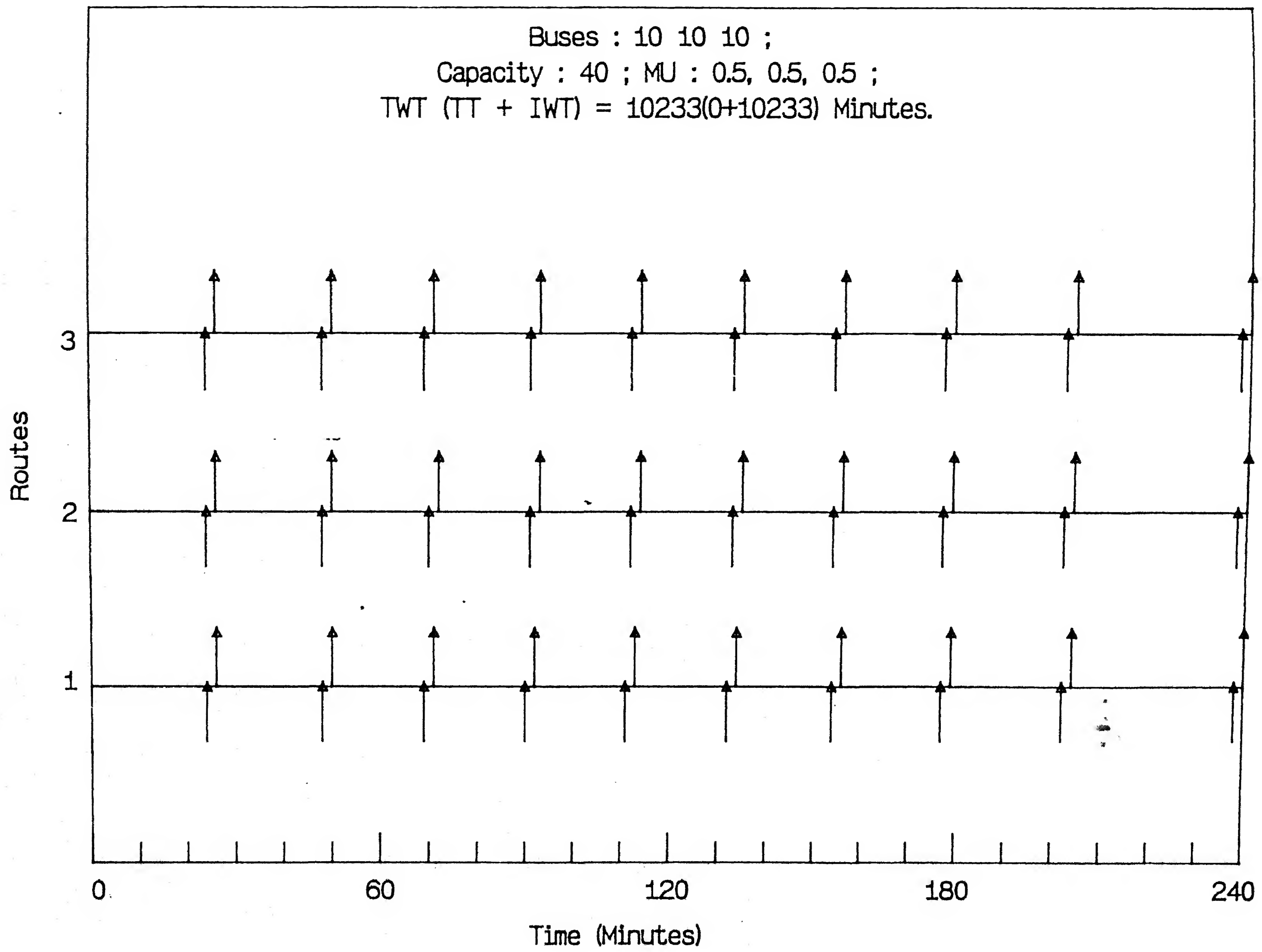

Figure 6.9: Optimal schedule for only IWT consideration with bus capacity 40 (non-uniform arrival pattern)

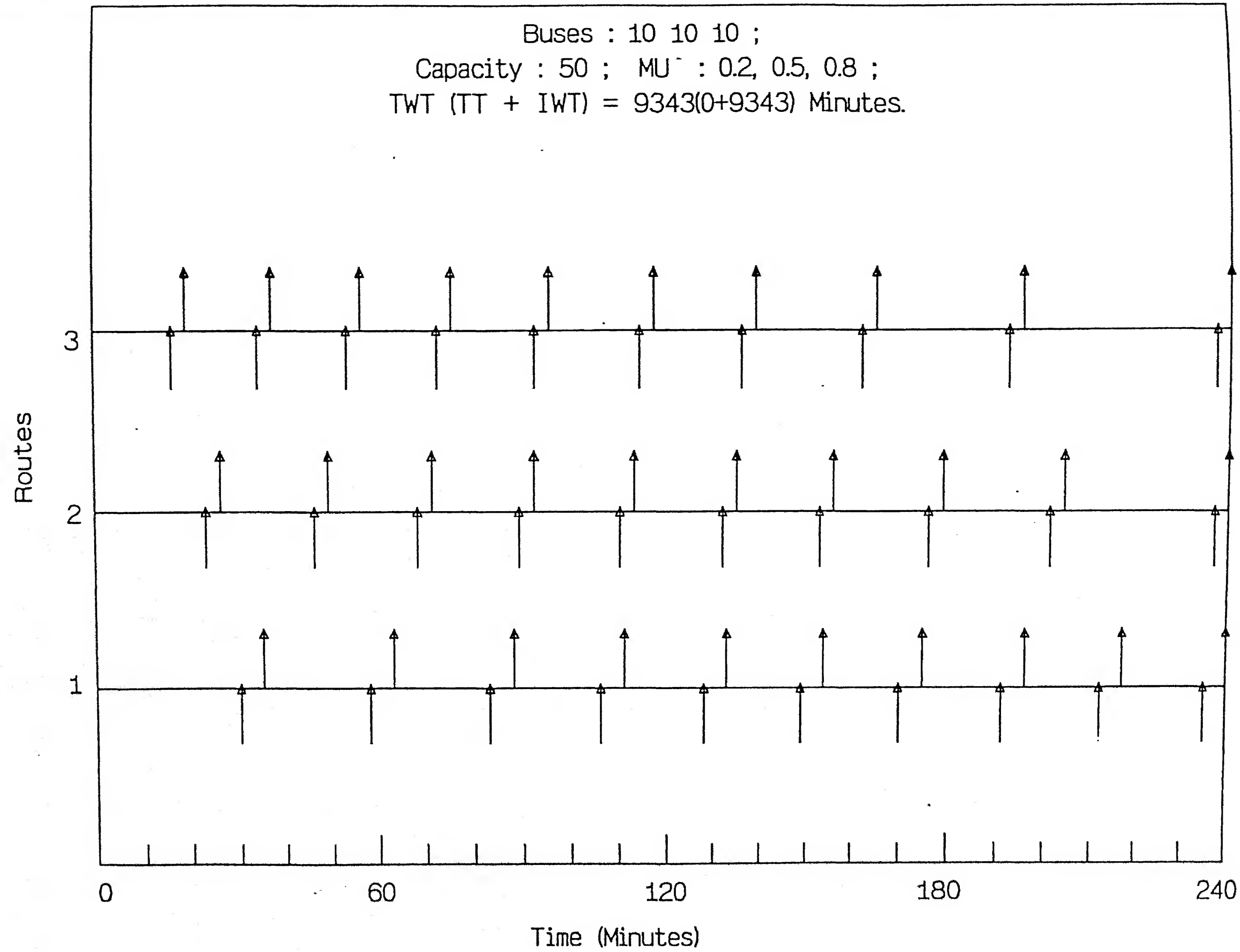

Figure 6.10: Optimal schedule for only IWT consideration with bus capacity 50 (non-uniform arrival pattern)

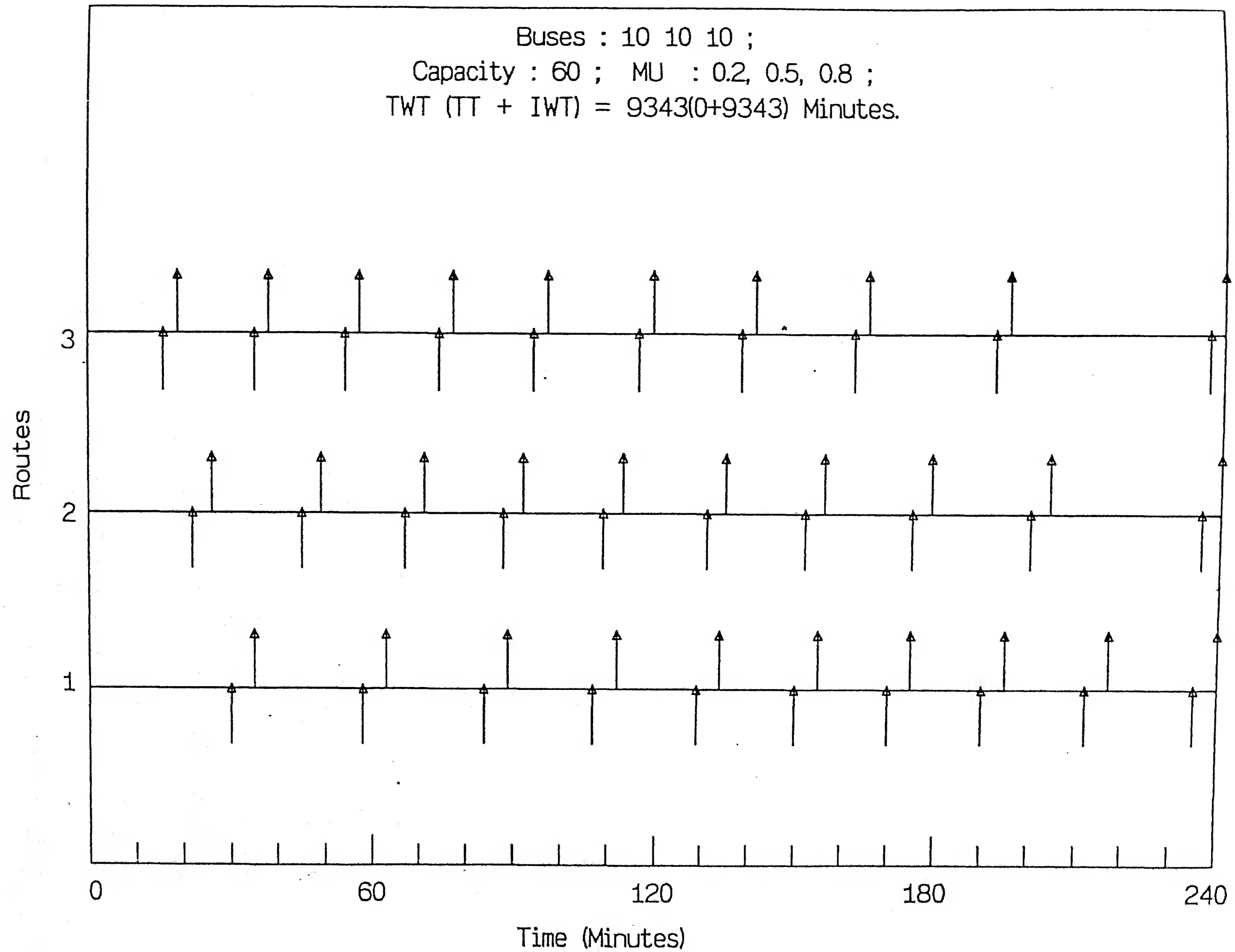

Figure 6.11: Optimal schedule for only IWT consideration with bus capacity 60 (non-uniform arrival pattern)

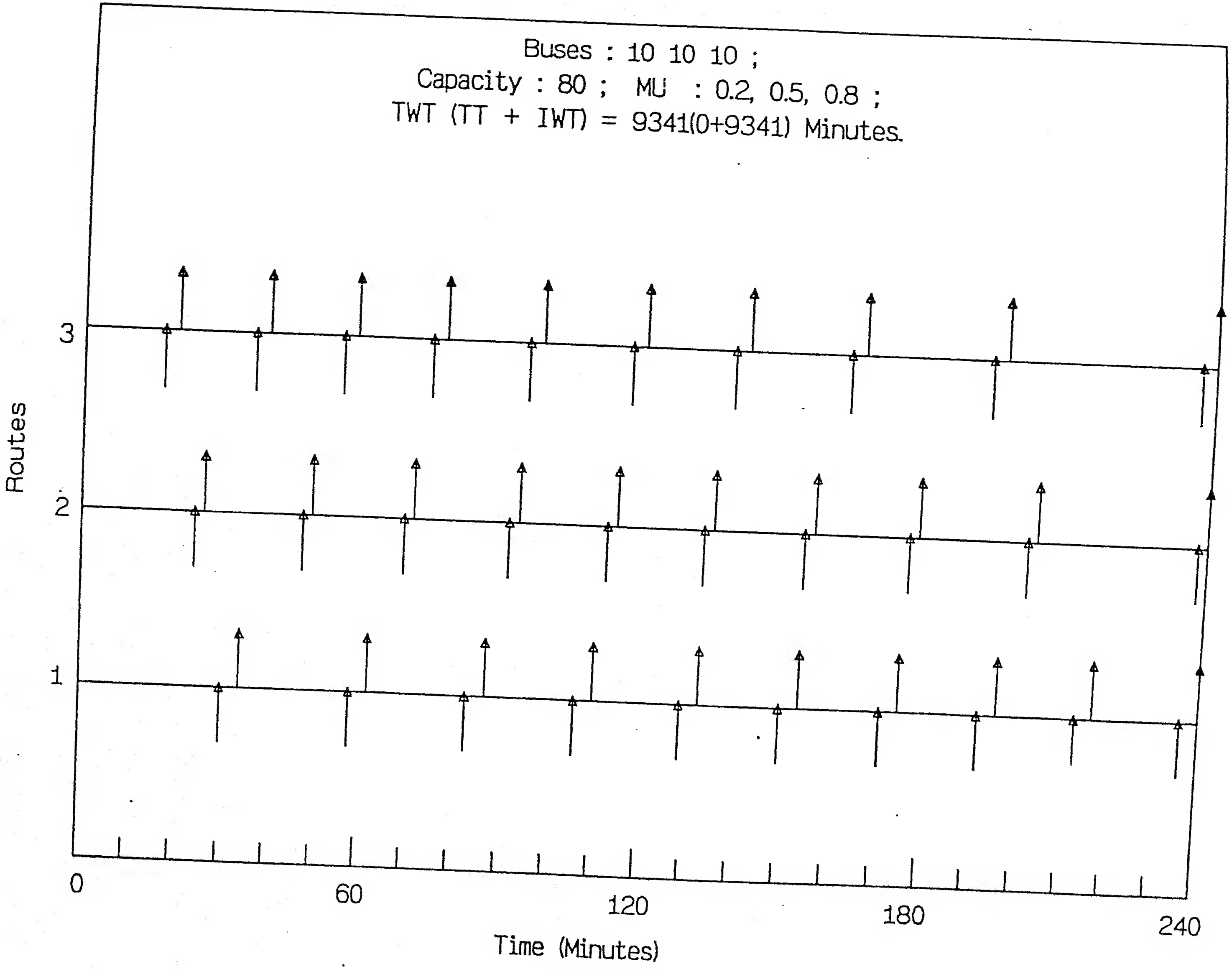

Figure 6.12: Optimal schedule for only IWT consideration with bus capacity 80 (non-uniform arrival pattern)

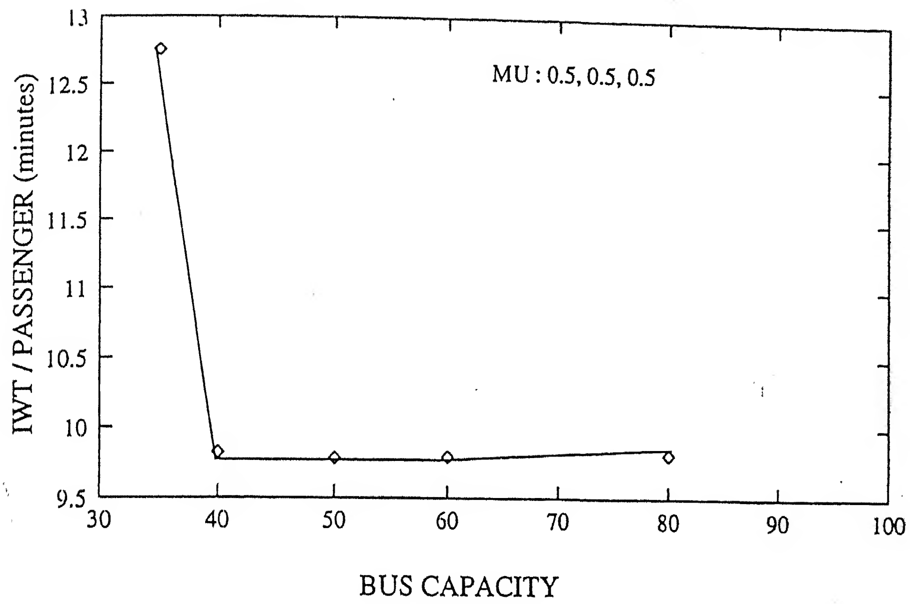

Figure 6.13: Graph for Bus Capacity versus IWT/passengers

Further one would expect the effect of bus capacity on IWT/passenger to be similar to that mentioned under Case II. Figure 6.13 shows this effect and all the observation made earlier remains, as expected applicable to this figure also.

6.2.4 Case IV

In this case both IWT and TT have been considered in the objective function. The μ value in $E_i(\tau)$ is assumed to be 0.2 and the η value in $W_{i,j}(\tau)$ is as mentioned earlier. In this case the total number of non-transferring passengers is approximately equal to 1060 and total number of transferring passengers is approximately equal to 100. Figure 6.14 and 6.15 represent the optimal schedule obtained under the above assumptions and with the assumption that the bus capacity is equal to 40. Figure 6.14 further assumes that $k = 1$ and Figure 6.15 assumes that $k = 10$; where k is the weight associated with transfer time. The purpose of using k is to allow the analyst to give more

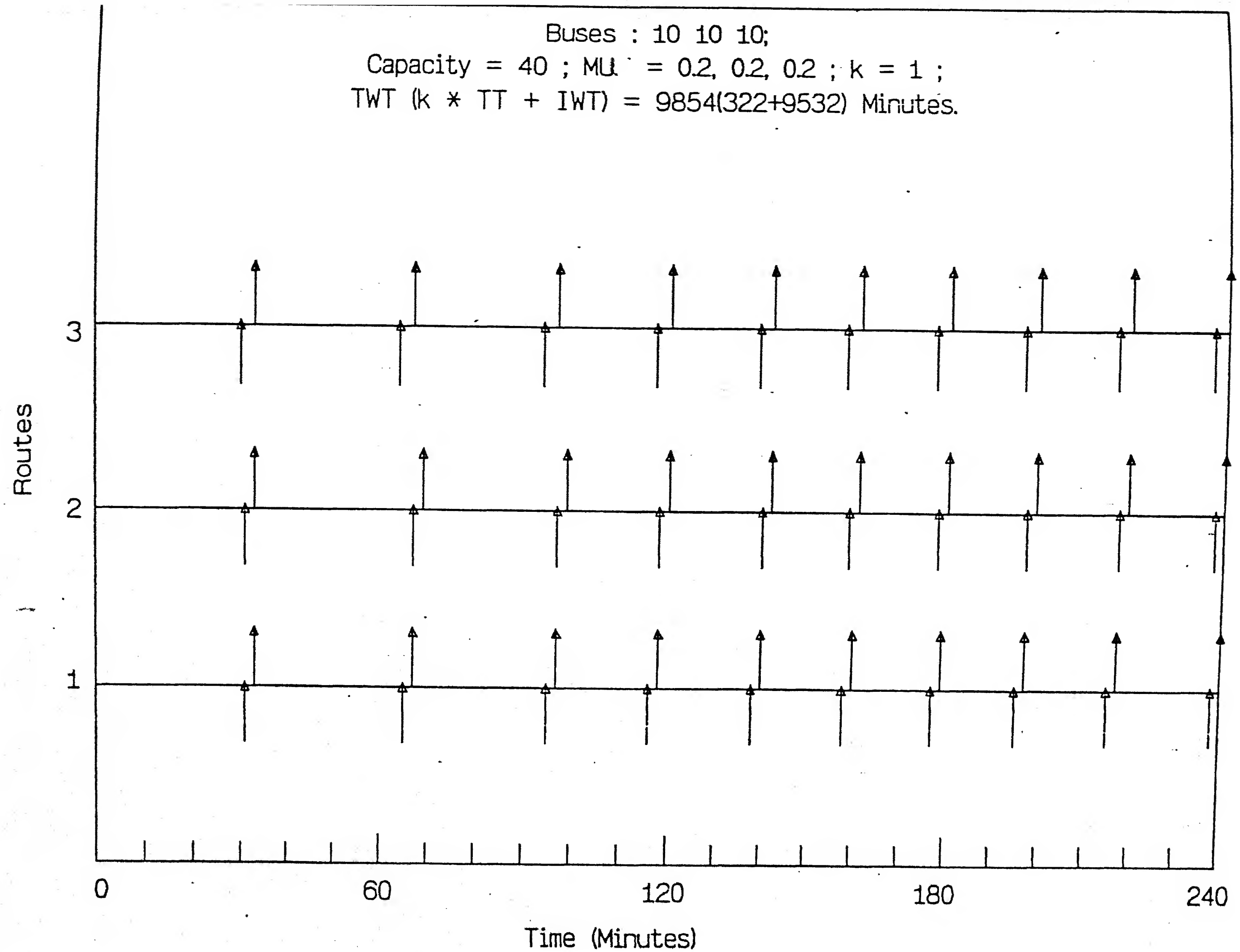

Figure 6.14: Optimal schedule for TWT consideration with bus capacity 40 and k = 1 (Non-uniform arrival pattern)

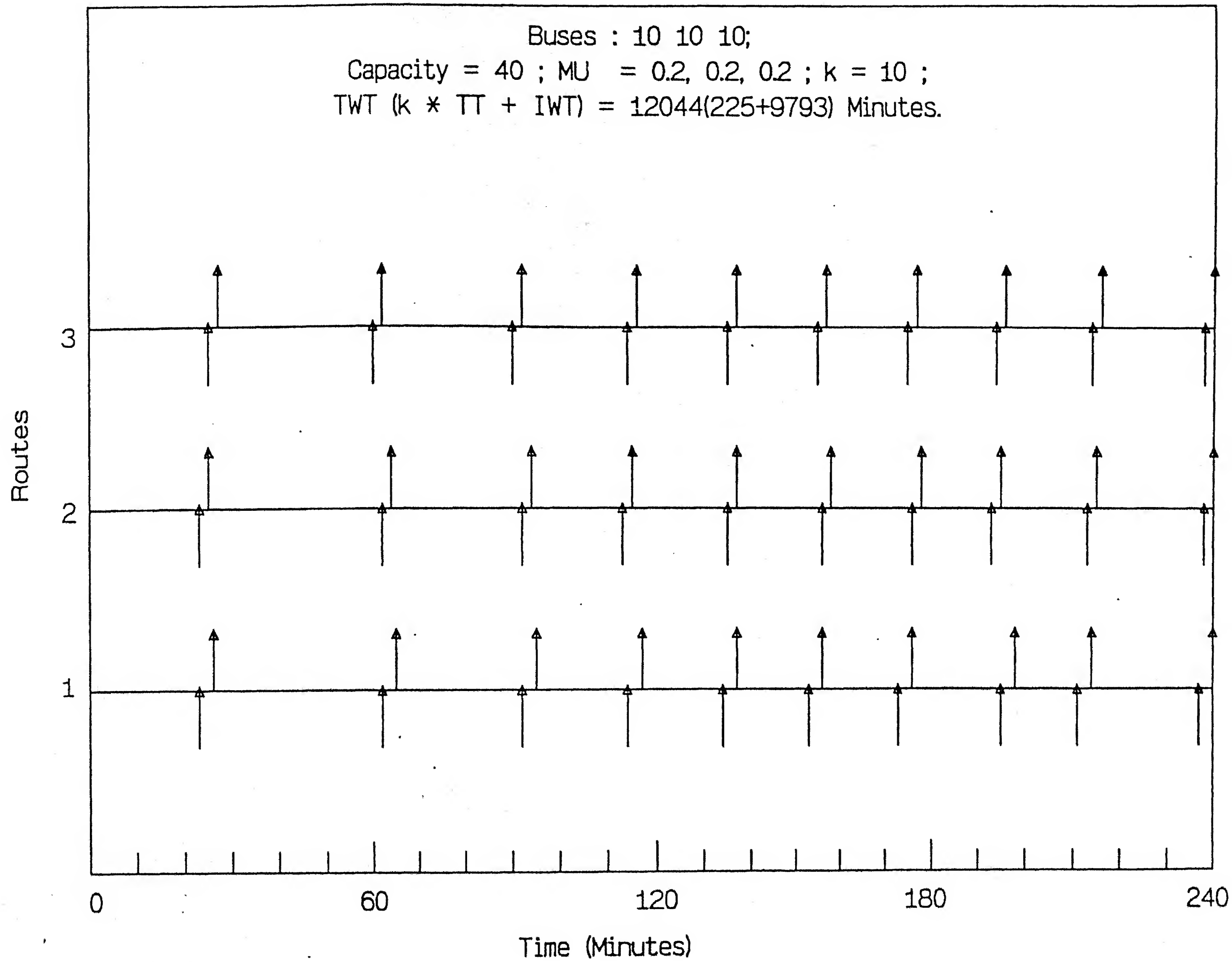

Figure 6.15: Optimal schedule for TWT consideration with bus capacity 40 and k = 10 (Non-uniform arrival pattern)

weight to each unit of transfer time then to each unit of initial waiting time. Both the figures show that the concentration of buses in the latter half of the scheduling time period than the former half. Further the buses on each route are aligned and the stopping times of buses are small. Under the above mention assumptions of parameter values the characteristics stated in the previous sentence are expected in an optimal schedule. Hence one can claim that the proposed methodology does give optimal/near optimal schedule.

Further as k increases from 1 to 10 one should expect a decrease in the transfer time per transferring passenger. This is indeed the case here ; when $k = 1$ the transfer time per transferring passenger is 3.2 minutes while the same is only 2.3 minutes when $k = 10$. Another positive point of algorithm which may be noted is that when transfer time per transferring passenger decreases, the initial waiting time per non-transferring passenger increases only to 10.2 minutes from 9.8 minutes. Figure 6.16 and 6.17 presents optimal schedules under the assumption that bus capacity is equal to 50 and $k = 1$ and 10, respectively. The other assumptions are same as those stated in this section. The general observations made for Figure 6.14 and 6.15 are also applicable here. Specifically, the transfer time per transferring passenger reduce from 2.4 minutes to 2.2 minutes when k increases from 1 to 10. The initial waiting time per non-transferring passenger remain more or less constant i.e., 9.8 minutes in both the figures.

6.2.5 Case V

The following parameter values have been assumed in this case. The value of μ in $E_i(\tau)$ is equal to 0.2 for route 1, 0.5 for route 2 and 0.8 for route 3. The value of η in $W_{i,j}(\tau)$ is same as mentioned earlier. The total number of non-transferring passengers is approximately equal to 1060 and the total number of transferring passengers is approximately equal to 100.

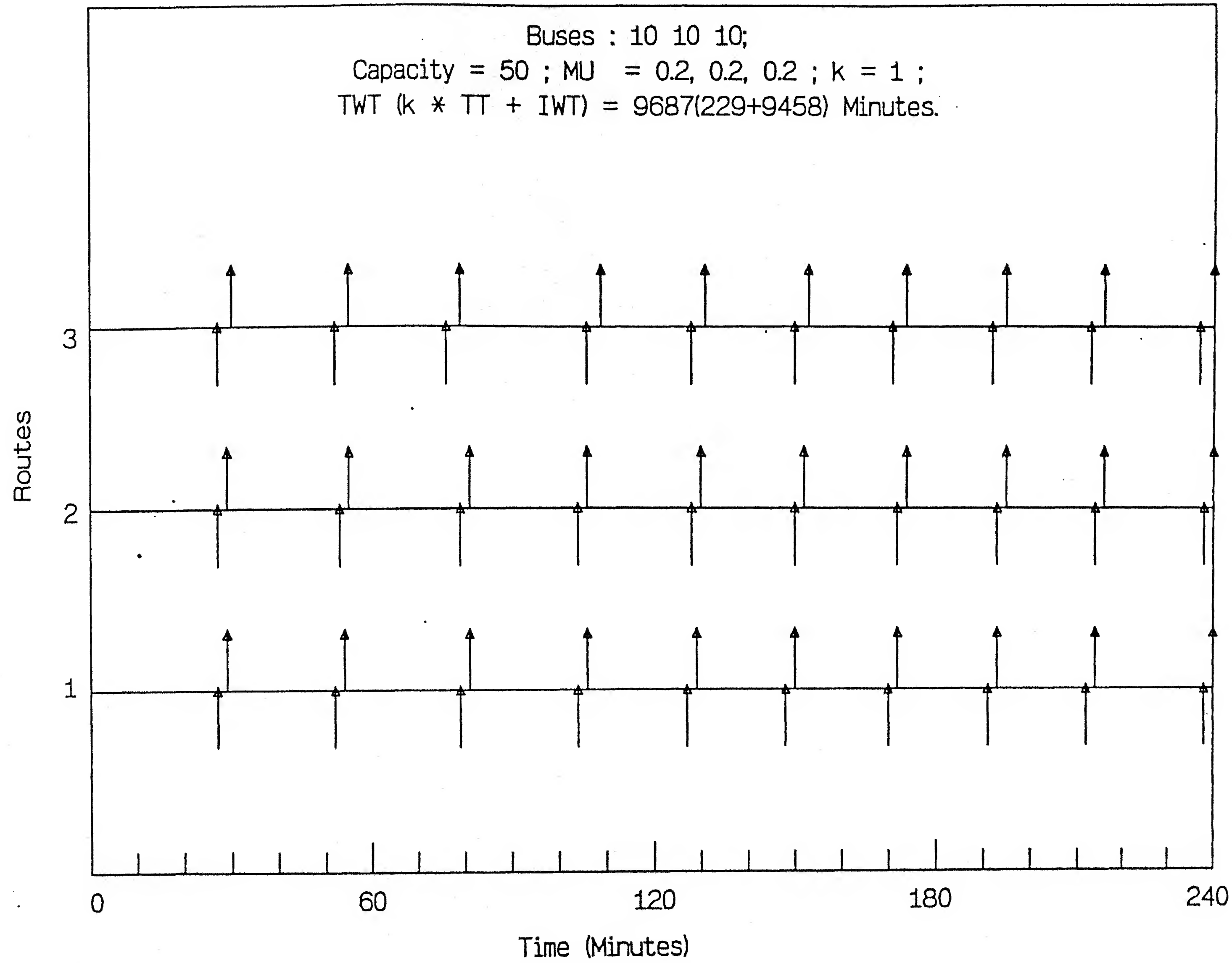

Figure 6.16: Optimal schedule for TWT consideration with bus capacity 50 and k = 1 (Non-uniform arrival pattern)

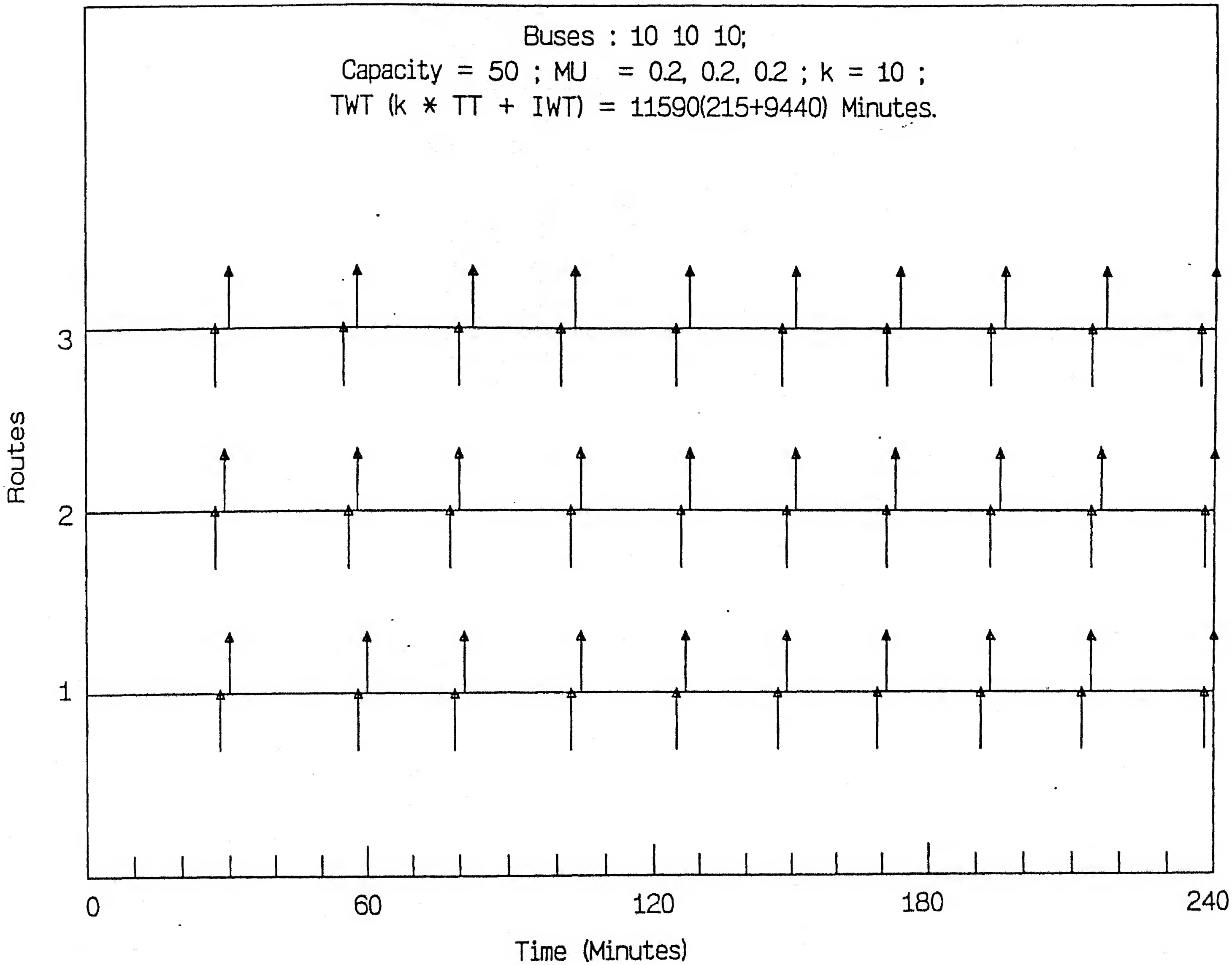

Figure 6.17: Optimal schedule for TWT consideration with bus capacity 50 and k = 10 (Non-uniform arrival pattern)

Figure 6.18 and 6.19 presents the best schedules obtained using the proposed methodology and under the assumption that the bus capacity is 40 and $k = 1$ and 10, respectively. It may be noted that determination of good schedule is difficult in this case because of the following reasons:
(a) The arrival pattern of non-transferring passengers is quite different from one route to other; for route 1 the arrival rate is the highest in the latter half of the scheduling time period, while for Route 2 it is highest around the middle of the scheduling period and for Route 3 the arrival pattern is the highest in the first half of the scheduling period. This means that from IWT standpoint the concentration of buses in Routes 1, 2 and 3 should be highest in the latter half, middle and former half of the scheduling period, respectively. (b) Yet, from the transfer time standpoint the buses for the various routes should be aligned. That is the optimality condition from IWT consideration and TT consideration are quite different and at conflict with one another.

The above facts are apparent from the schedules. For example (a) the stopping time of buses are larger than in the corresponding scenarios presented in the previous case; (b) the number of buses which are aligned (i.e., buses of different routes whose sojourns at the station at least partially overlapped) are much less in these cases then the corresponding scenarios of the earlier case, and (c) the total transfer time is quite large (9.7 minutes when $k = 1$ and 6.8 when $k = 10$). The interesting features to note here are (i) initial waiting time for non-transferring passenger is quite comparable to earlier scenarios and (ii) as k increases the transfer time per transferring passenger reduces substantially (obviously the number of alignments also increases substantially).

As the capacity of buses increases the possibility of the existence of schedules which offer lesser waiting time increases (as discussed earlier). The proposed algorithm does enfact find better schedules when capacity increases to 50 (see Figure 6.20 and 6.21).

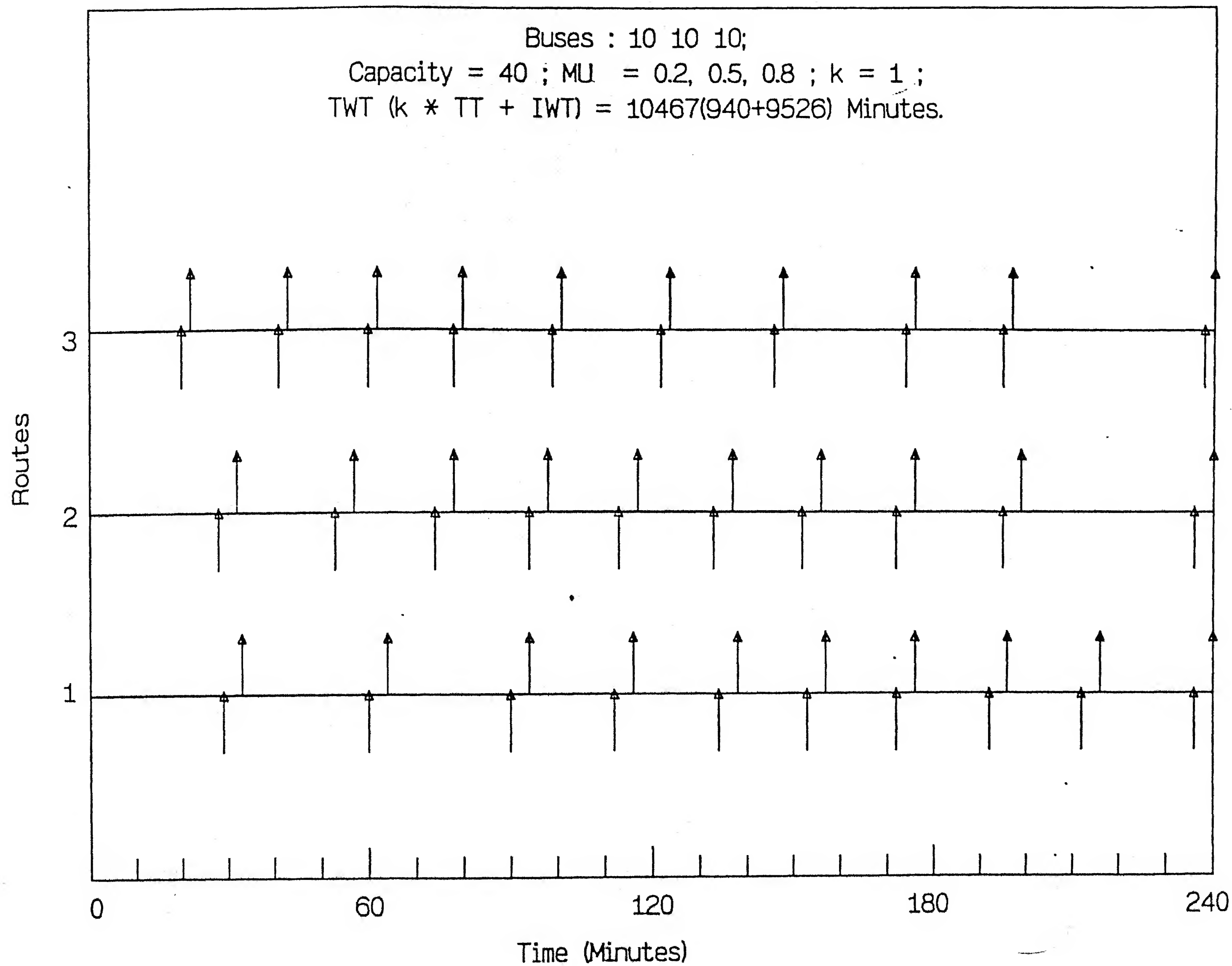

Figure 6.18: Optimal schedule for TWT consideration with bus capacity 40 and k = 1 (Non-uniform arrival pattern)

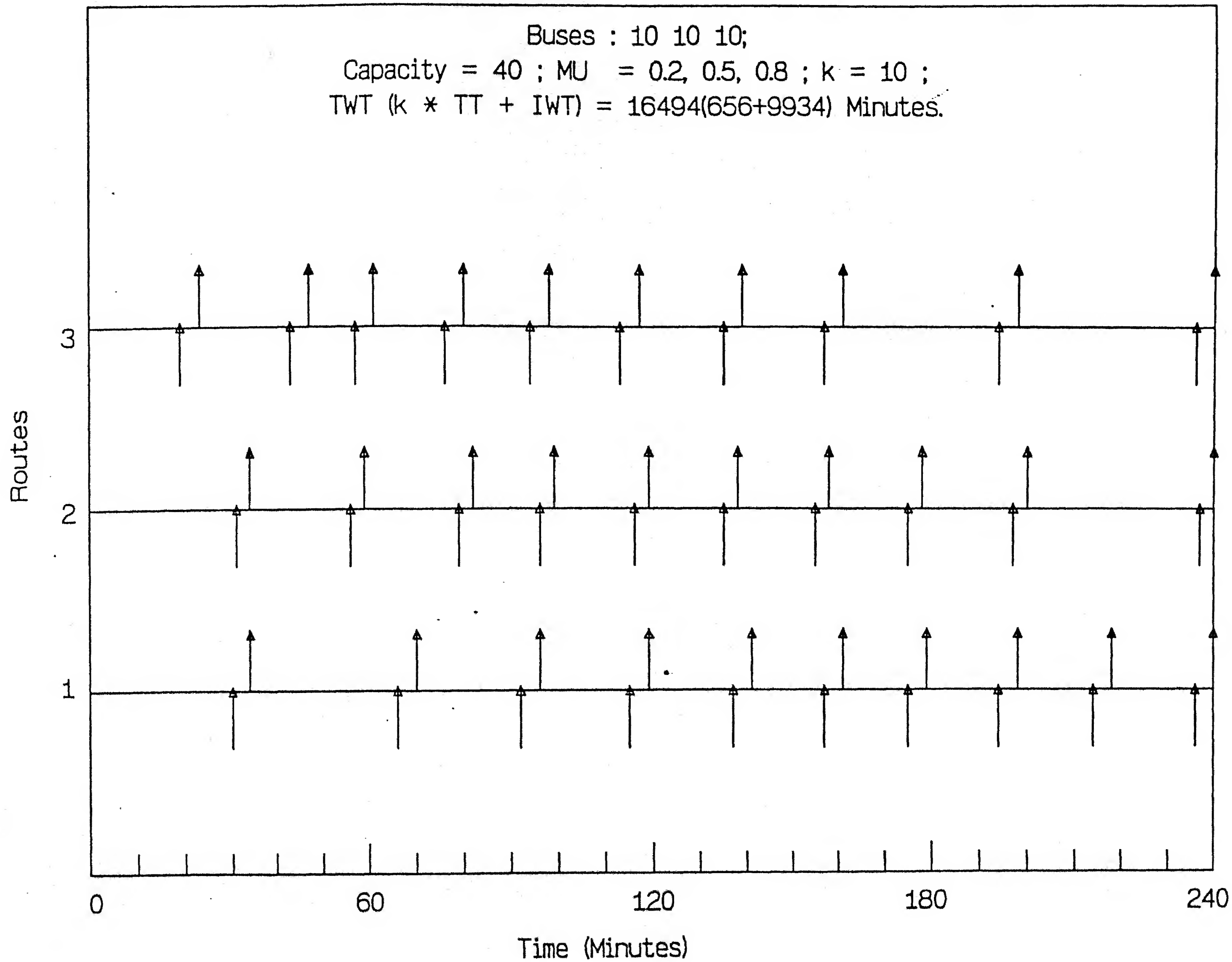

Figure 6.19: Optimal schedule for TWT consideration with bus capacity 40 and k = 10 (Non-uniform arrival pattern)

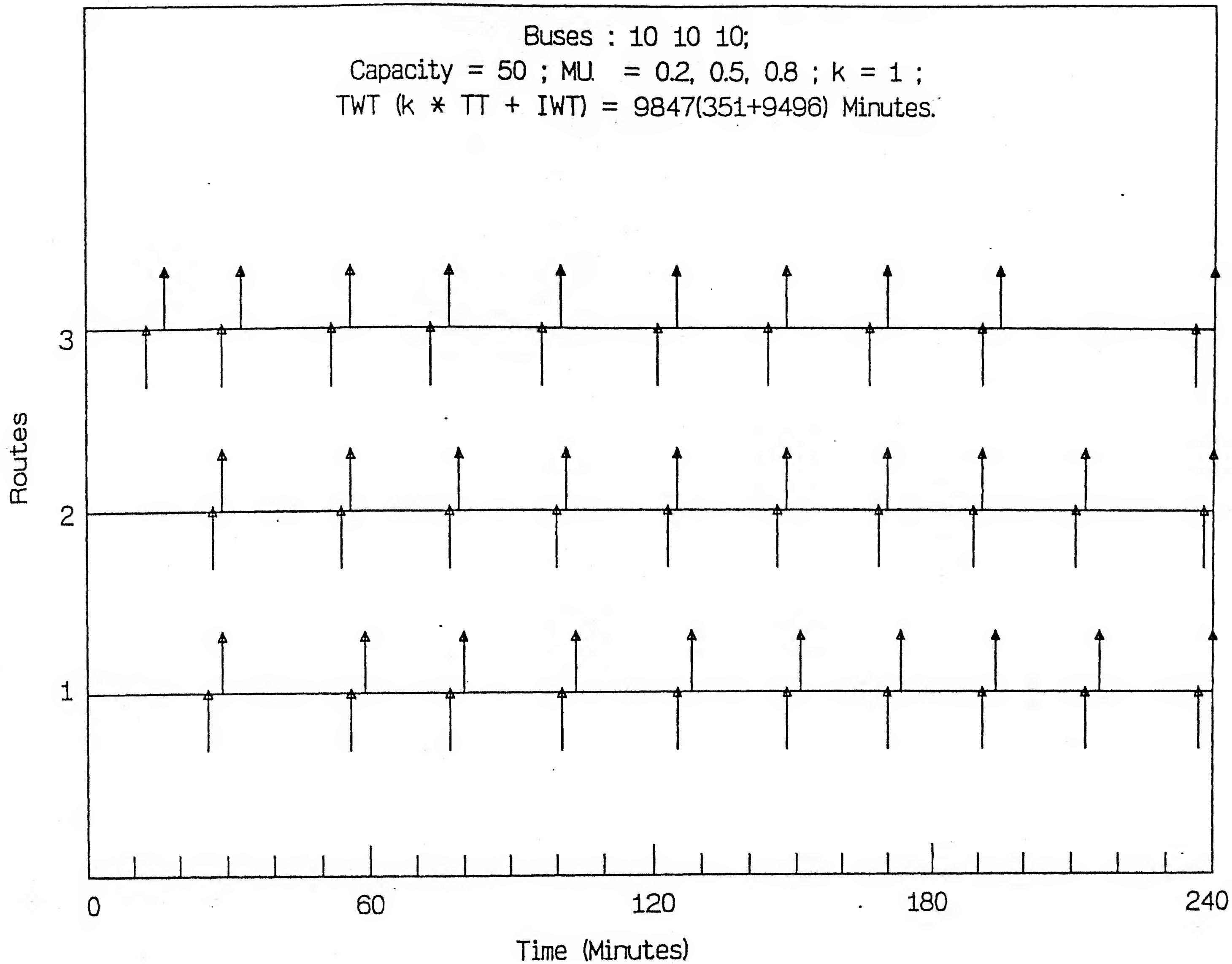

Figure 6.20: Optimal schedule for TWT consideration with bus capacity 50 and k = 1 (Non-uniform arrival pattern)

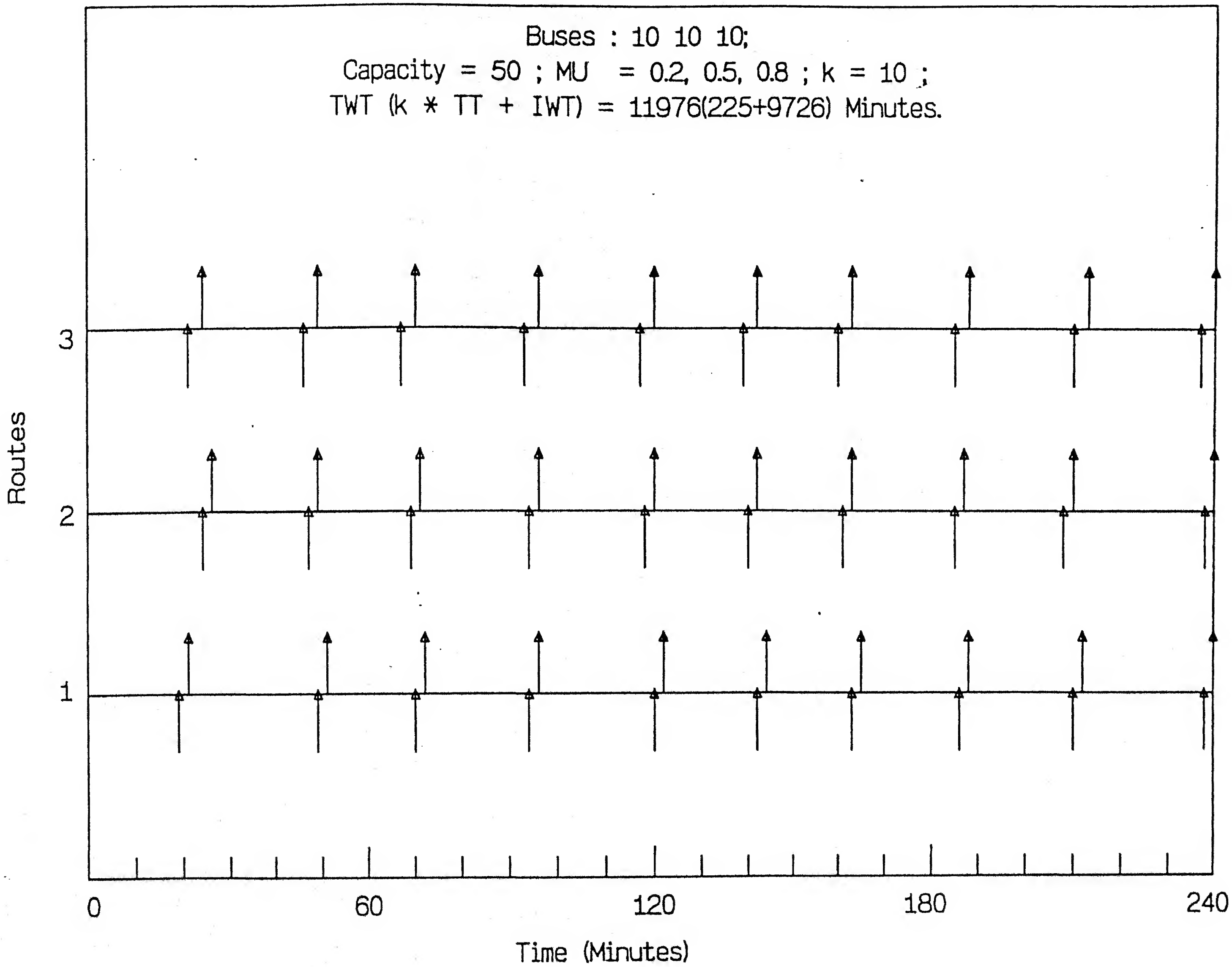

Figure 6.21: Optimal schedule for TWT consideration with bus capacity 50 and k = 10 (Non-uniform arrival pattern)

The transfer time per transferring passenger is 3.7 (for $k = 1$) and 2.3 minutes (for $k = 10$) and the initial waiting time per non-transferring passenger is 9.8 (for $k = 1$) and 10.1 minutes (for $k = 10$). These values are comparable to the corresponding scenarios of the previous case, despite the greater difficulty of obtaining good schedules in this case. These facts show that the proposed methodology is quite effective in obtaining optimal schedules even in difficult problem scenarios.

6.2.6 Case VI

This case is very similar to case IV except that total number of transferring passengers is approximately equal to 200. Figure 6.22 and 6.23 show the optimal schedules obtained assuming a bus capacity of 50. The initial waiting time per passenger is equal to 10.0 minutes for both case $k = 1$ and $k = 10$. The transfer time per passenger, however, reduces from 2.6 minutes to 2.5 minutes when k increases from 1 to 10. These results are comparable to the results obtained in similar scenarios present in Case IV. The fact that the results are comparable even when the total number of transferring passengers is double again shows the competence of proposed algorithm in finding optimal schedules.

Figure 6.24 and 6.25 present optimal schedules for the same assumptions of parameter values but with the bus capacity of 80. The initial waiting time per non-transferring passenger for $k = 1$ and $k = 10$ are 9.9 and 1.9 minutes and 10.2 and 1.8 minutes, respectively. The reduction in transfer time per transferring passenger could be achieved because of larger bus capacity.

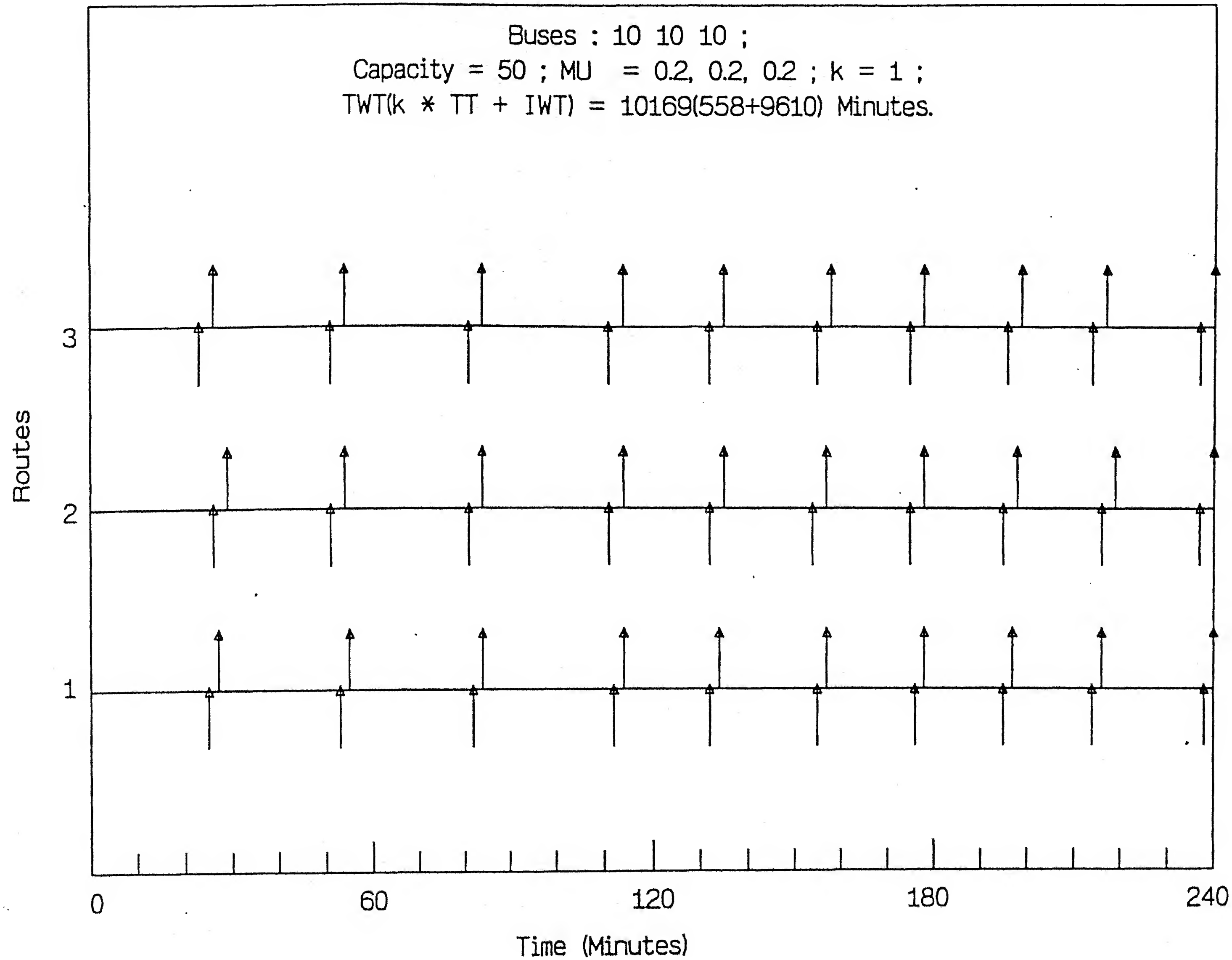

Figure 6.22: Optimal schedule for TWT consideration with bus capacity 50, k = 1 and transferring passengers ≈ 200 (Non-uniform arrival pattern)

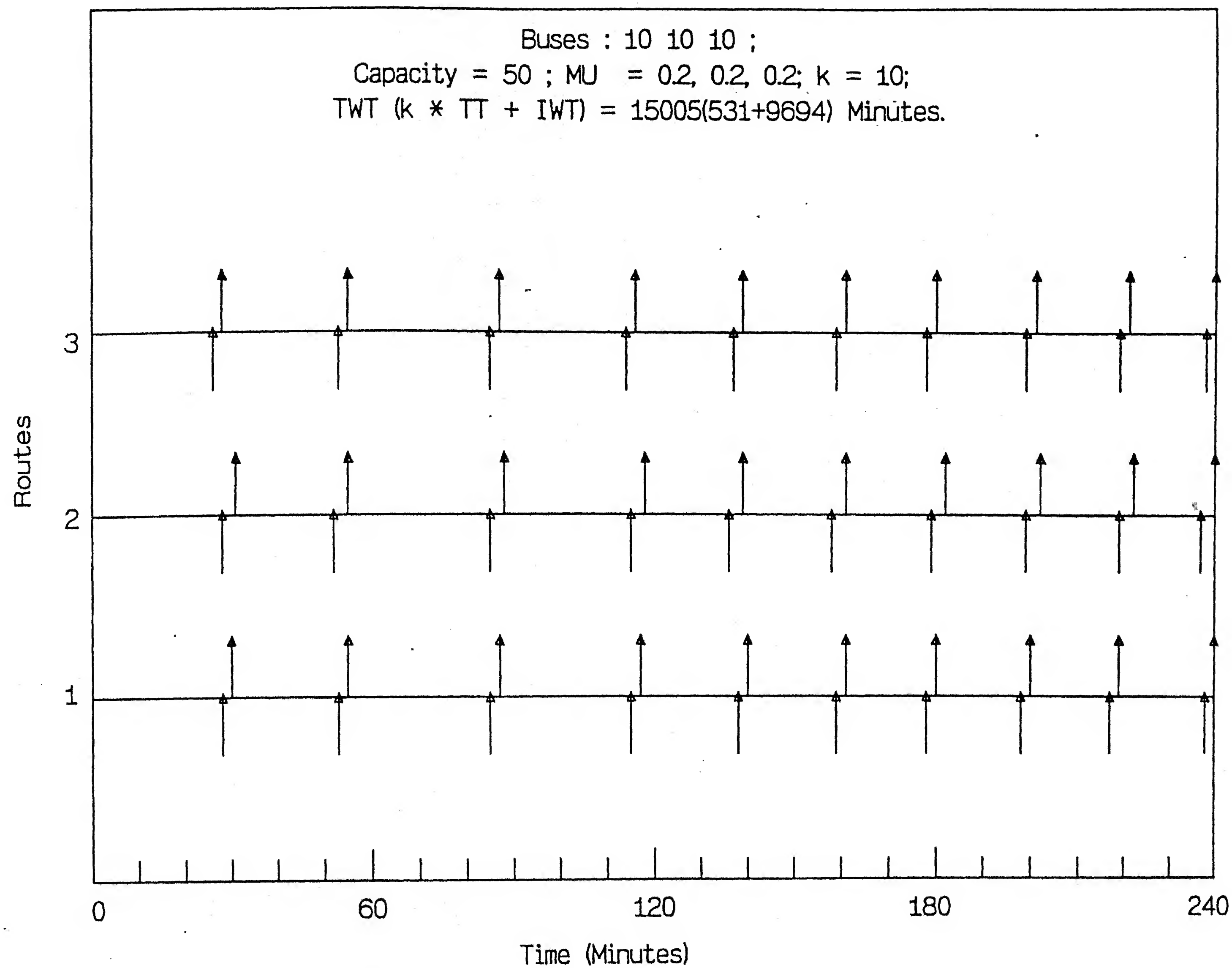

Figure 6.23: Optimal schedule for TWT consideration with bus capacity 50, k = 10 and transferring passengers ≈ 200 (Non-uniform arrival pattern)

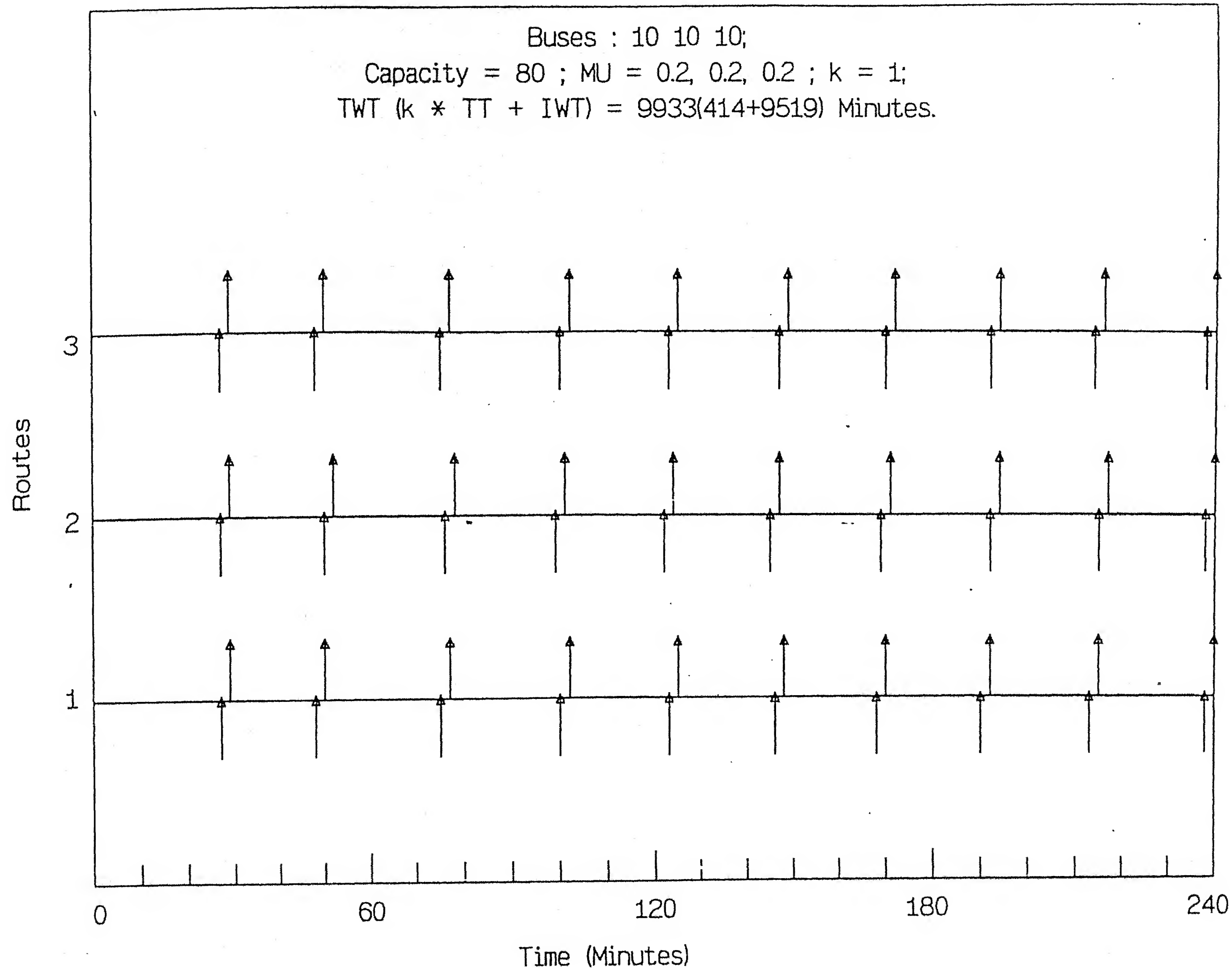

Figure 6.24: Optimal schedule for TWT consideration with bus capacity 80, $k = 1$ and transferring passengers ≈ 200 (Non-uniform arrival pattern)

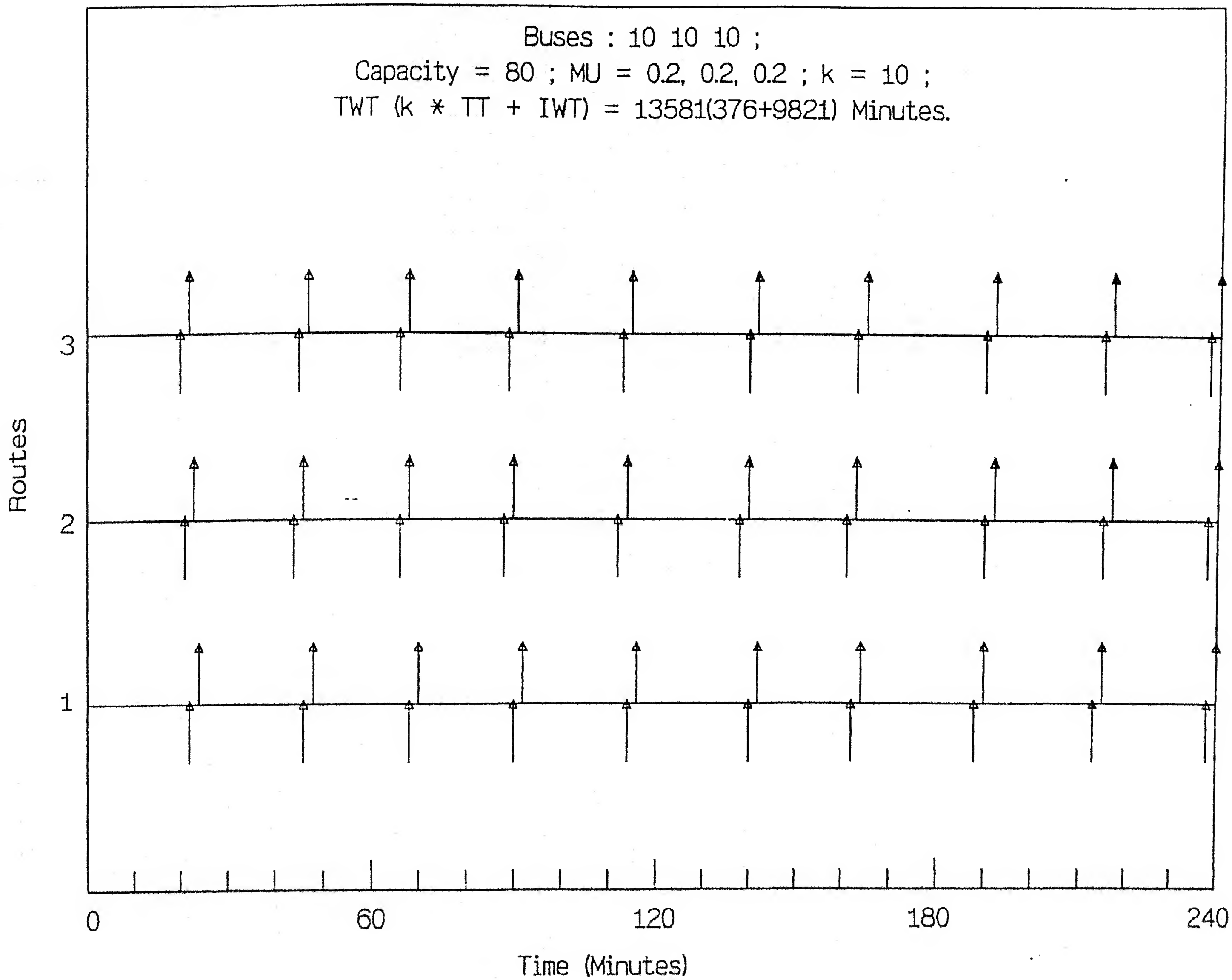

Figure 6.25: Optimal schedule for TWT consideration with bus capacity 80, $k = 10$ and transferring passengers $\approx$ 200 (Non-uniform arrival pattern)

6.2.7 Case VII

This case is similar to Case V except that the total number of transferring passengers are approximately 200. As in Case VI bus capacities of 50 and 80 are used. Figure 6.26 and 6.27 show the optimal schedule when the bus capacity is 50 and Figure 6.28 and Figure 6.29 show the optimal schedules when bus capacity is 80. The initial waiting time per non-transferring passenger for each of the schedule is approximately 10.0 minutes. When the bus capacity is 50 the transfer time per transferring passenger reduces from 5.3 minutes to 4.5 minutes as k increases from 1 to 10. These values are somewhat greater than those corresponding scenarios in Case V (Figure 6.20 and 6.21). The increase in the transfer time per transferring passenger could be attributed to the effect of more passengers at the station combined with a conflicting arrival pattern.

However, when bus capacity is increased to 80 the transfer time per transferring passenger is 2.2 minutes when $k = 1$ and 1.8 minutes when $k = 10$. These values are obviously better than those obtain for Case V i.e., when the effect of the increased demand is offset by as increase in bus capacity, the transfer time per transferring passenger becomes comparable (if not better) than those obtained in Case V.

In earlier cases were devised to test the proposed procedure for cases where the optimal schedules can be reasonably well predicted (for example when arrival patterns are similar and number of buses on each route are equal). It was shown that the best schedules obtained using the proposed algorithms did match the expected optimal schedules in the above cases.

Hence, after gaining confidence in the proposed algorithm, the same is

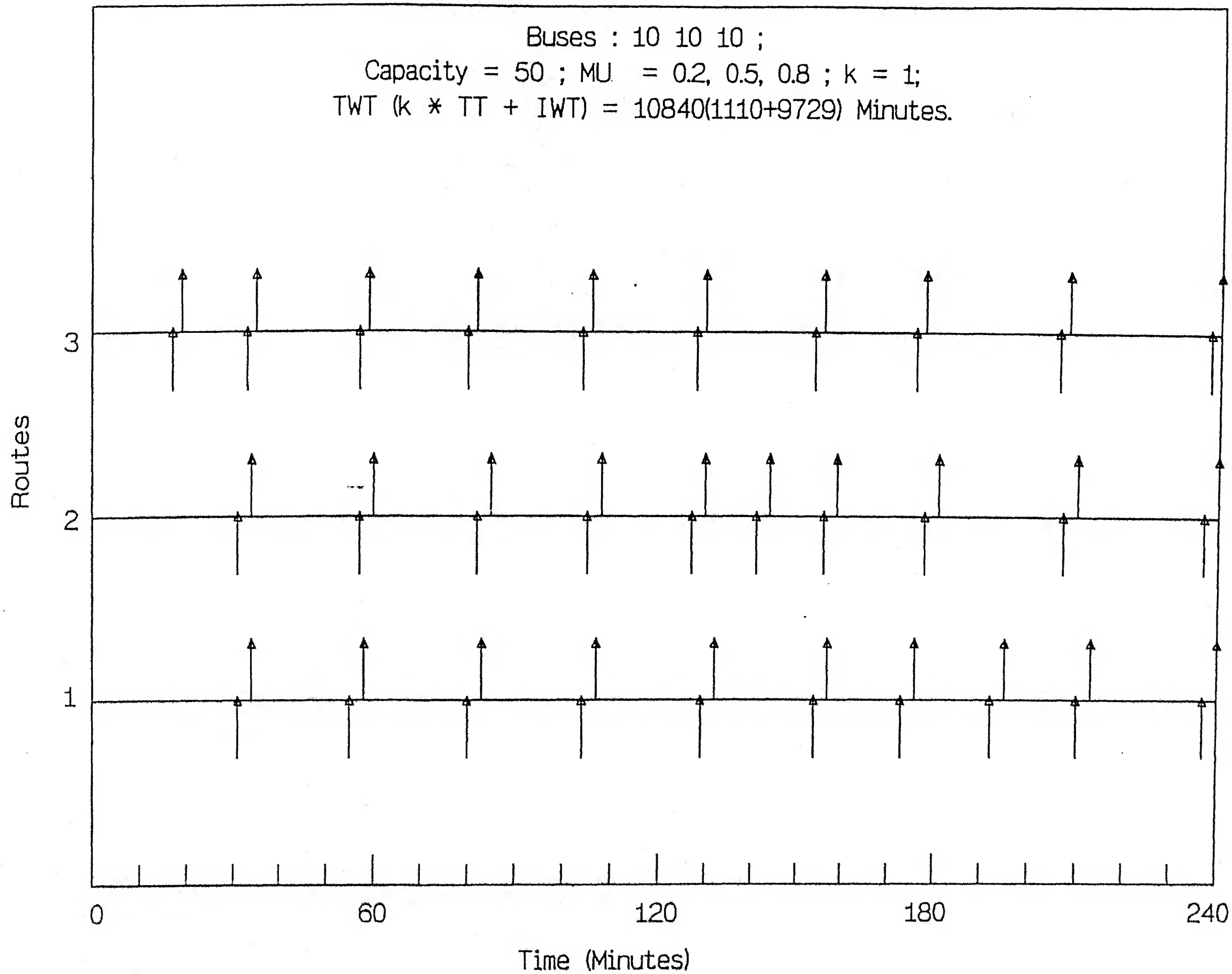

Figure 6.26: Optimal schedule for TWT consideration with bus capacity 50, k = 1 and transferring passengers ≈ 200 (Non-uniform arrival pattern)

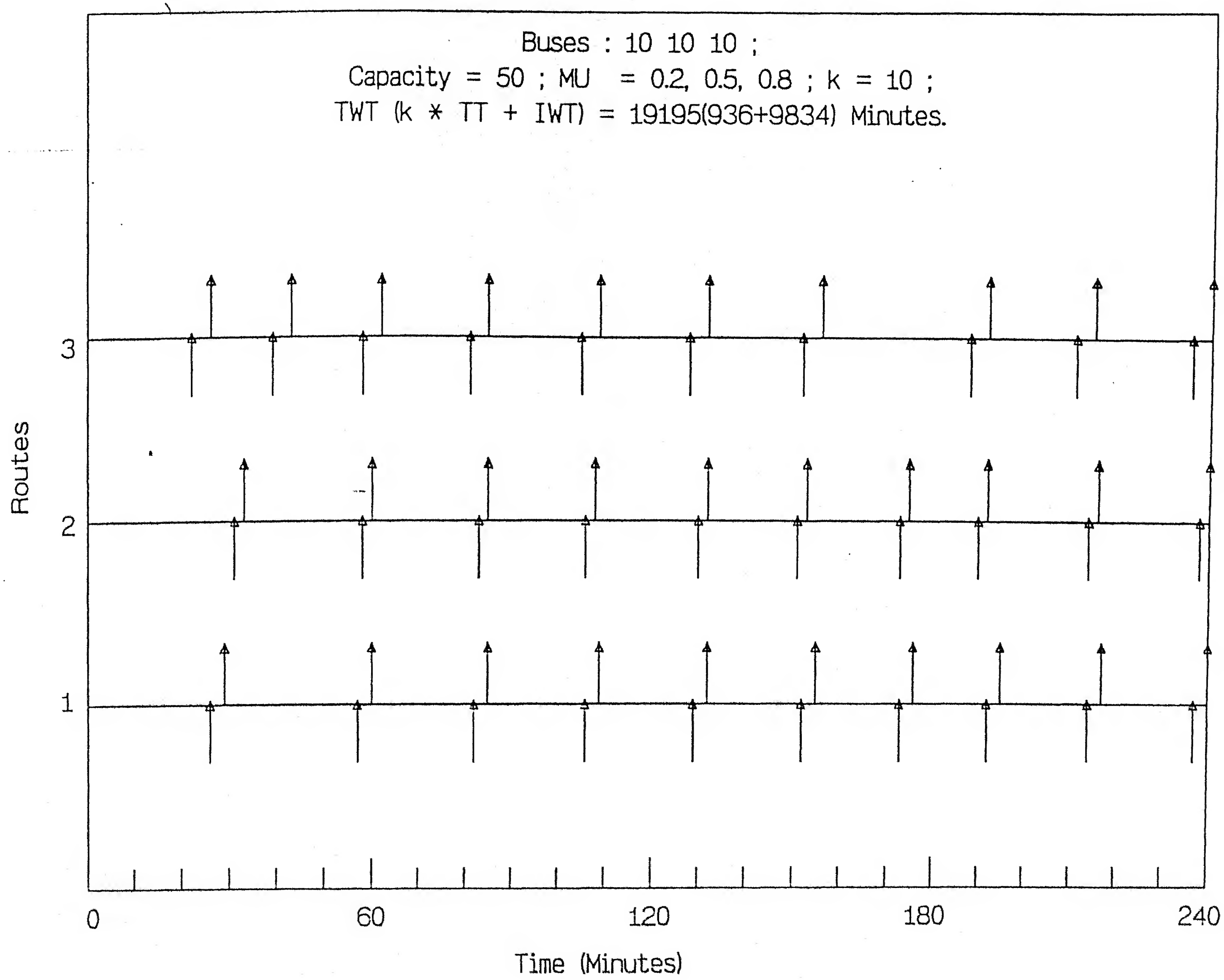

Figure 6.27: Optimal schedule for TWT consideration with bus capacity 50, k = 10 and transferring passengers ≈ 200 (Non-uniform arrival pattern)

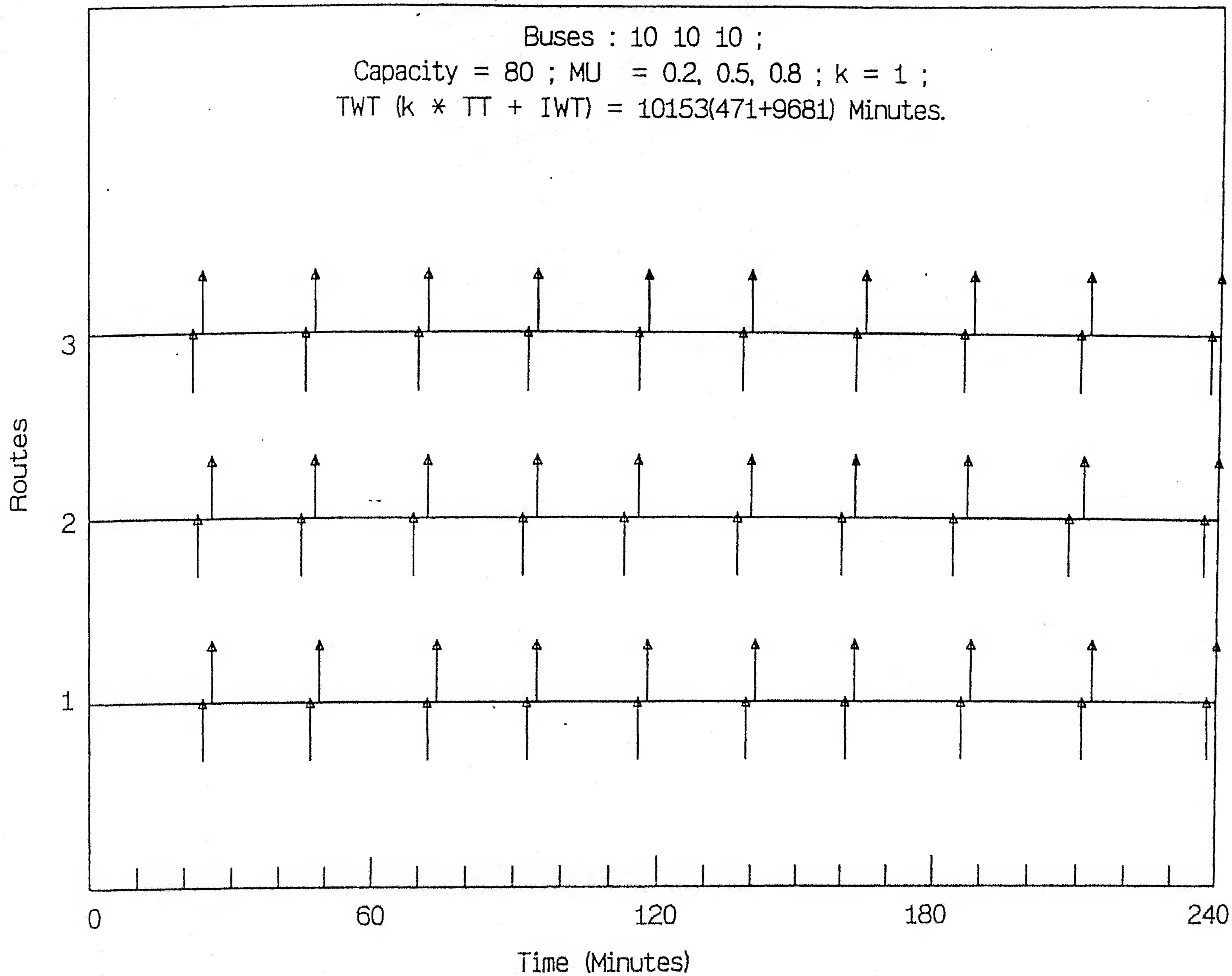

Figure 6.28: Optimal schedule for TWT consideration with bus capacity 80, $k = 1$ and transferring passengers ≈ 200 (Non-uniform arrival pattern)

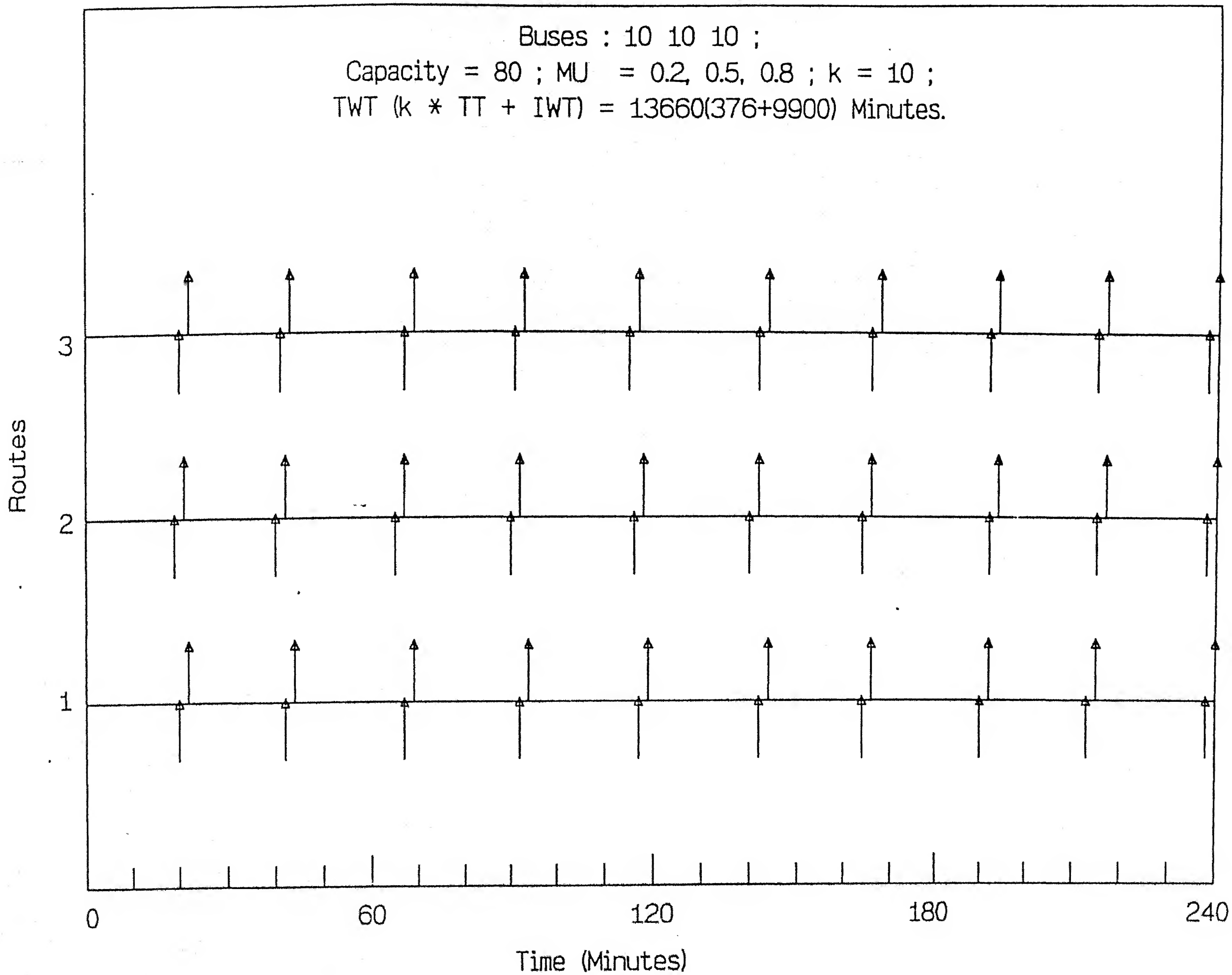

Figure 6.29: Optimal schedule for TWT consideration with bus capacity 80, k = 10 and transferring passengers ≈ 200 (Non-uniform arrival pattern)

tested on following two cases. It may be noted that in these two cases the number of buses on each route is different (8 buses on Route 1, 10 buses on Route 2 and 12 buses on Route 3) and thus one cannot identify the nature of the optimal schedules. These cases have been provided in order to show that the algorithm can handle general version (as opposed to equal resources on all routes) of the scheduling problem, as is expected (since, the algorithm does not make any assumptions on the resource allocation between the routes).

6.2.8 Case VIII

In this case optimal schedules are presented assuming μ to be 0.2 for all routes and η to be as mentioned earlier. The total number of transferring passengers is approximately 100 and total number of non-transferring passengers is approximately 1050. Optimal schedules for bus capacity of 50 are presented in Figures 6.30 (for $k = 1$) and 6.31 (for $k = 10$). Optimal schedules with bus capacity of 80 are shown in Figure 6.32 (for $k = 1$) and 6.33 (for $k = 10$).

On studying and comparing Figures 6.30 and 6.31 the following features become apparent : (i) the stopping time reduce as k increases (recall that transfer times are affected by the stopping times), (ii) the number of times the buses on all the routes are aligned increase from 6 to 7 as k increases (note that from the definition of "aligned" a maximum of 8 is possible in these cases), (iii) the alignments are much clearer when $k = 10$ than when $k = 1$, and (iv) the concentration of buses in all the routes is greater towards the latter part of the scheduling period. It may be noted that one can also observe the effect of k by comparing the transfer time per transferring passenger, it is 7.5 minutes when $k = 1$ and 5.7 minutes when $k = 10$.

On studying and comparing Figures 6.32 and 6.33 one can see, as earlier, the stopping times reduce and alignments increase (from 6 to 8) when k

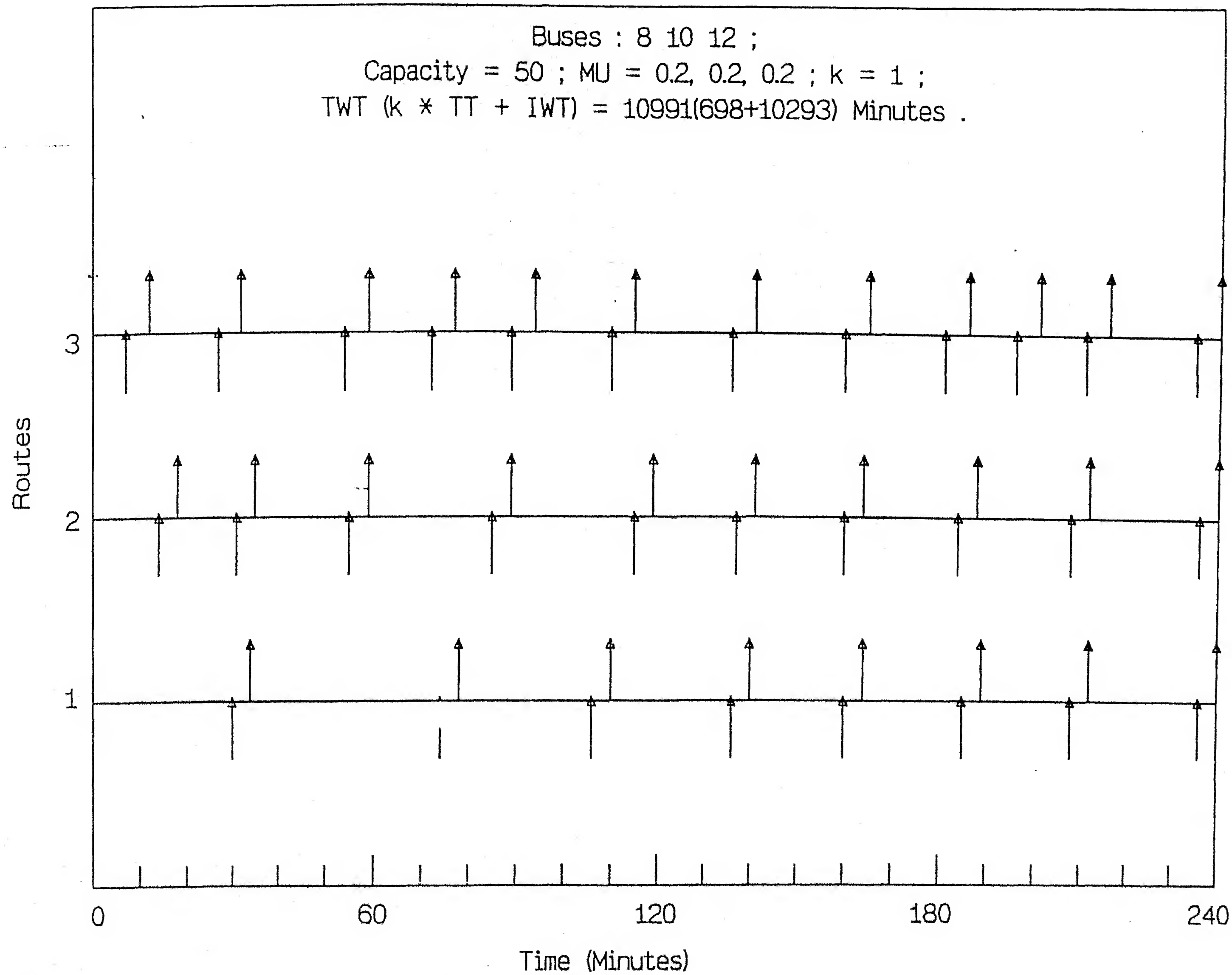

Figure 6.30: Optimal schedule for TWT consideration with bus capacity 50 and k = 1 (Non-uniform arrival pattern and unequal resources)

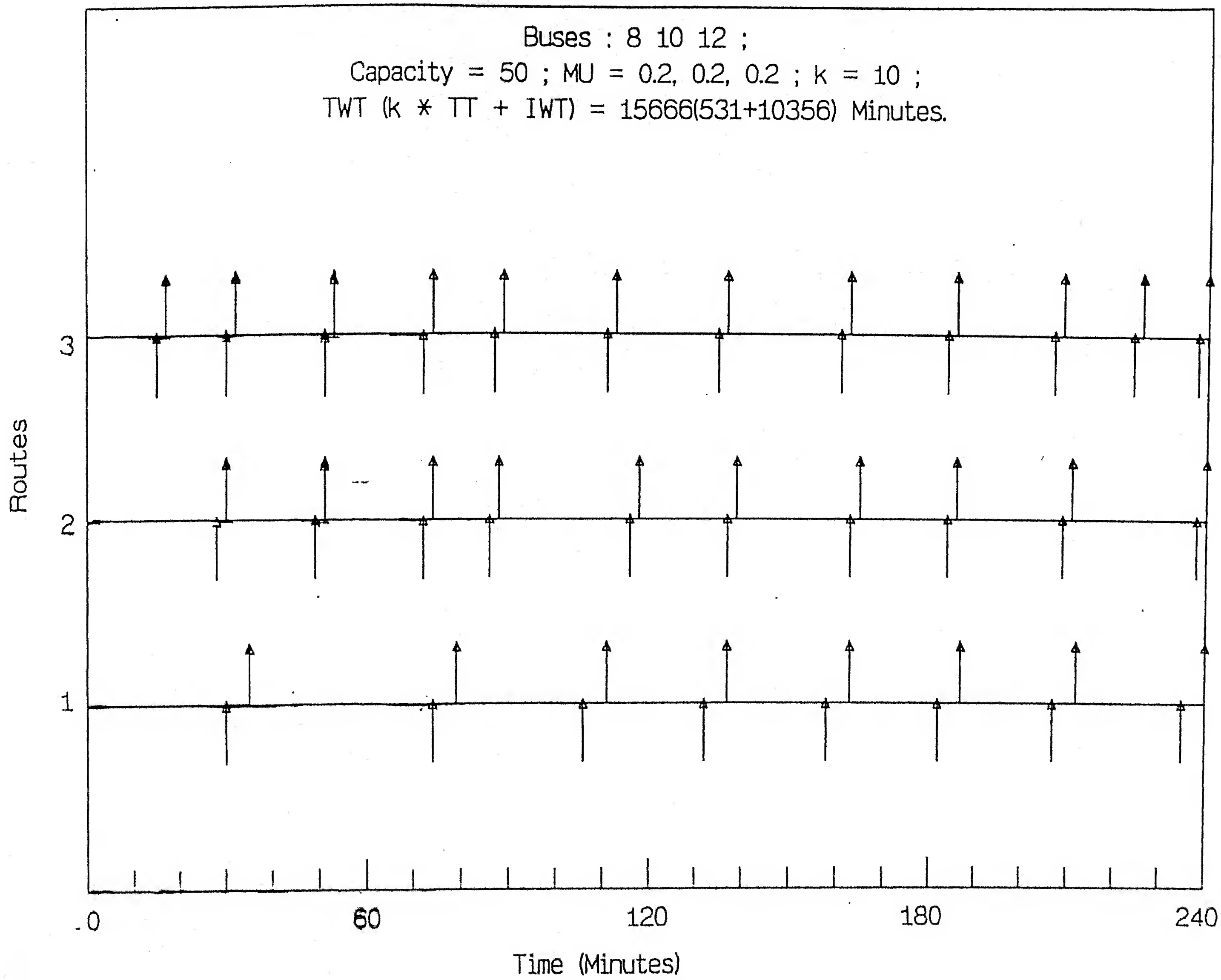

Figure 6.31: Optimal schedule for TWT consideration with bus capacity 50 and k = 10 (Non-uniform arrival pattern and unequal resources)

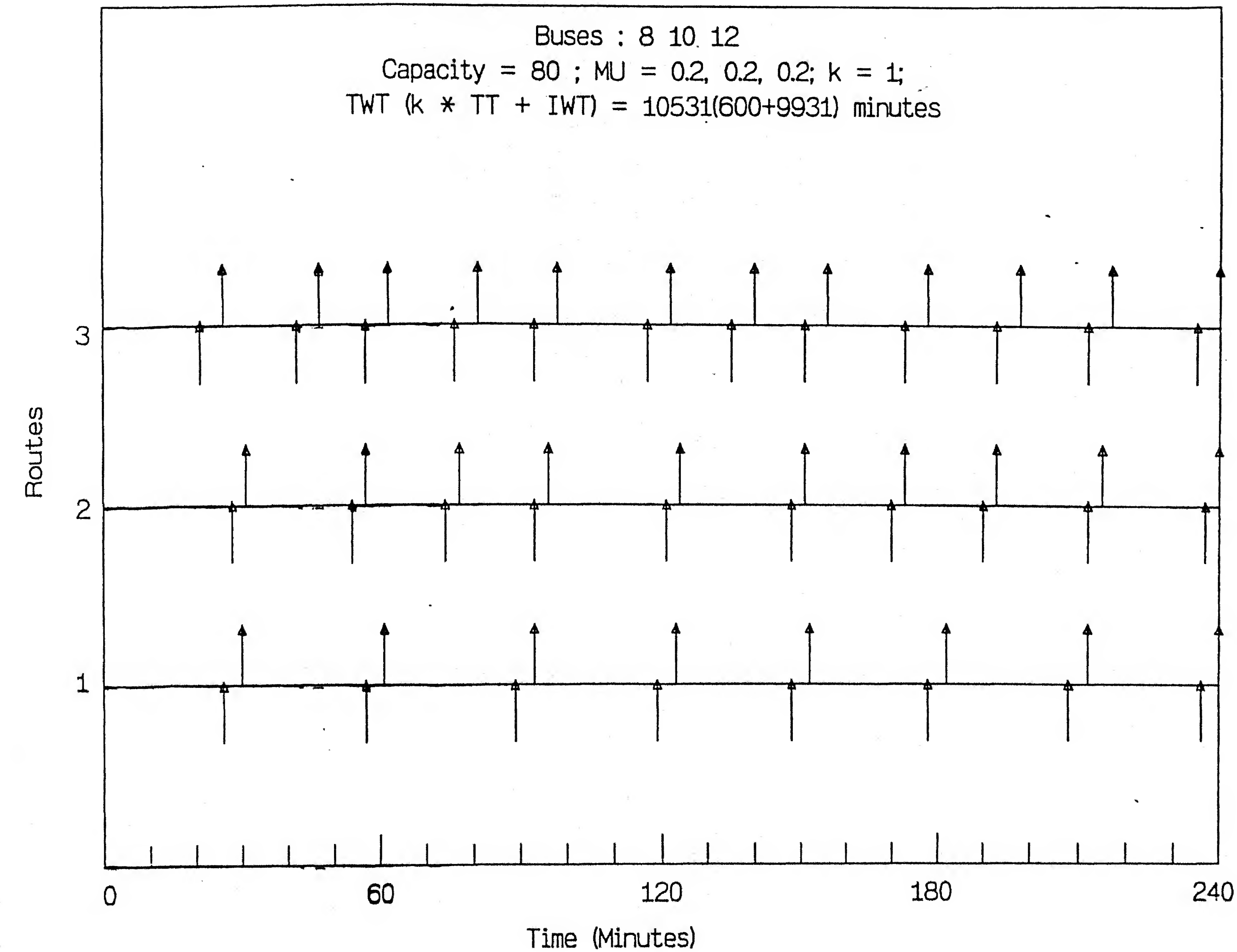

Figure 6.32: Optimal schedule for TWT consideration with bus capacity 80 and k = 1 (Non-uniform arrival pattern and unequal resources)

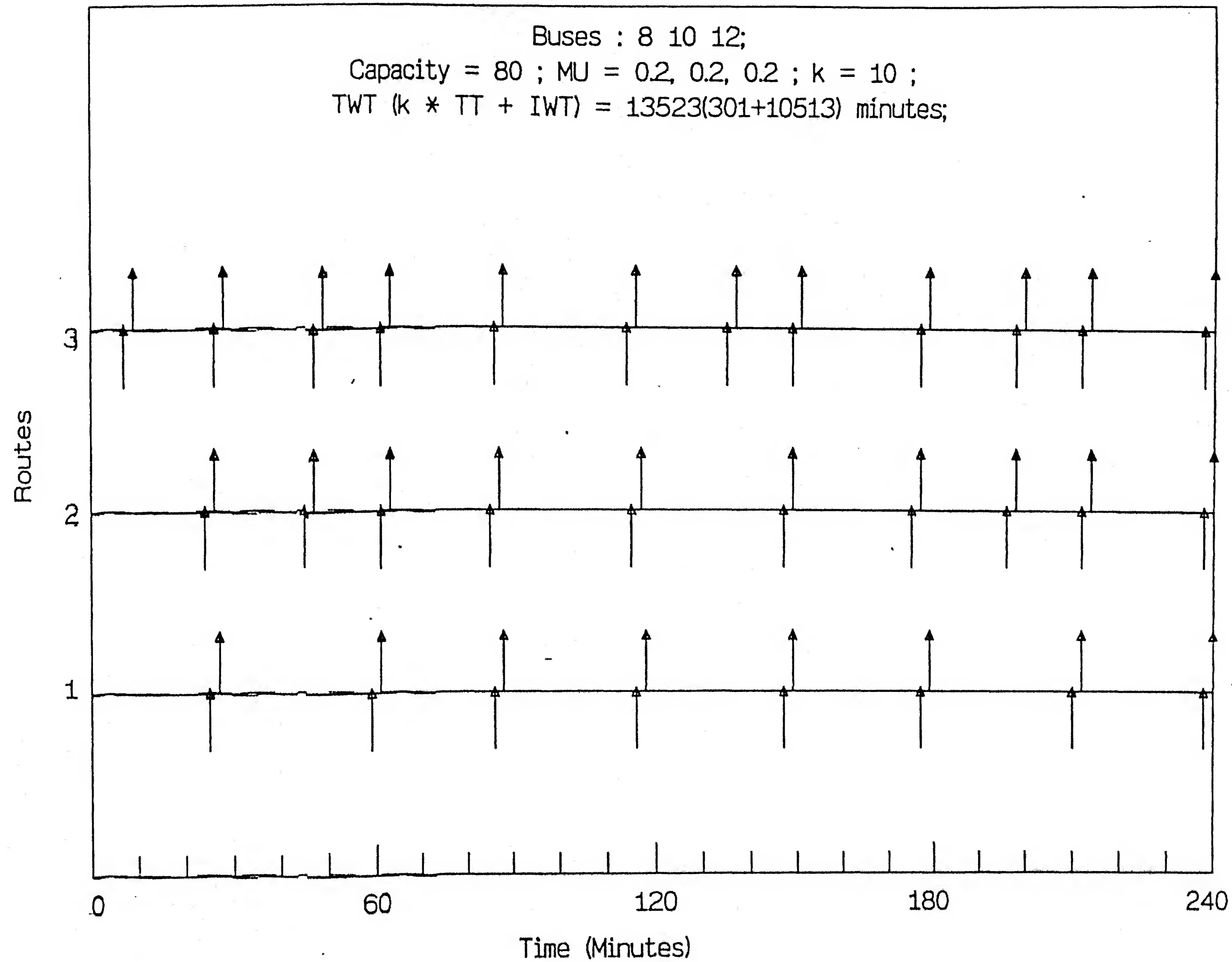

Figure 6.33: Optimal schedule for TWT consideration with bus capacity 80 and k = 10 (Non-uniform arrival pattern and unequal resources)

increases. The interesting observation that can be made here is that a substantial decrement in transfer time per transferring passenger (when compared to the schedules with capacity equal to 50) can be achieved due to the added capacity (6.2 minutes when $k = 1$ and 3.4 minutes when $k = 10$).

Thus, although unlike earlier, one cannot compare the best schedules obtained here with a priori expecting of the optimal schedules, one can still suggest that the best schedules obtained here do posses features which could qualify these as optimal/near-optimal schedules.

6.2.9 Case IX

Although various other test cases were studied, this is the last Case for which results are being presented here. The assumptions of this case are : (a) μ in $E_1(\tau) = 0.2$, μ in $E_2(\tau) = 0.5$ and μ in $E_3(\tau) = 0.8$, (b) η as mentioned earlier, (c) total number of transferring passengers is approximately 100 and that of non-transferring passengers are approximately 1050, and (d) bus capacity is equal to 80.

The best schedules for $k = 1$ and $k = 10$ are presented in Figures 6.34 and 6.35, respectively. Expected behavior with respect to k and μ values are also observed here. For example, the number of "alignments" increase from 6 to 8 and transfer time per transferring passenger reduce from 5.4 to 3.5 minutes as k increases; and generally the concentration of buses follows the arrival pattern of non-transferring passengers. It is interesting to note the tradeoff between TT and IWT as k increases. In Figure 6.34, 7 out of 12 buses on Route 3 arrive in the first half (since $\mu = 0.8$ means more non-transferring arrive in the first half) yet in Figure 6.35 the number is reduced to 6, possible in order to reduce the transfer time by increasing "alignments" (since a larger value of k means a greater weight to TT in the objective function).

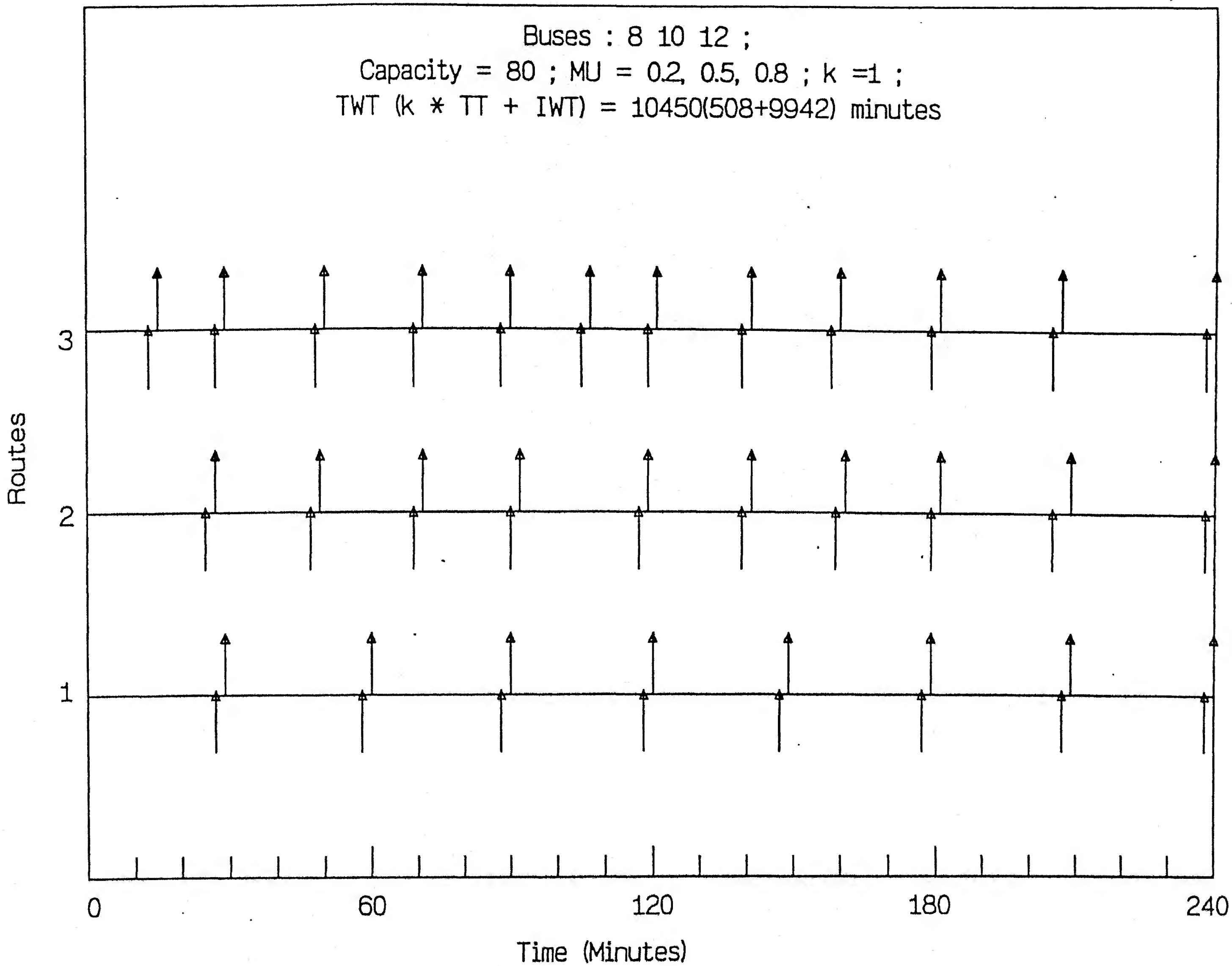

Figure 6.34: Optimal schedule for TWT consideration with bus capacity 80 and k = 1 (Non-uniform arrival pattern and unequal resources)

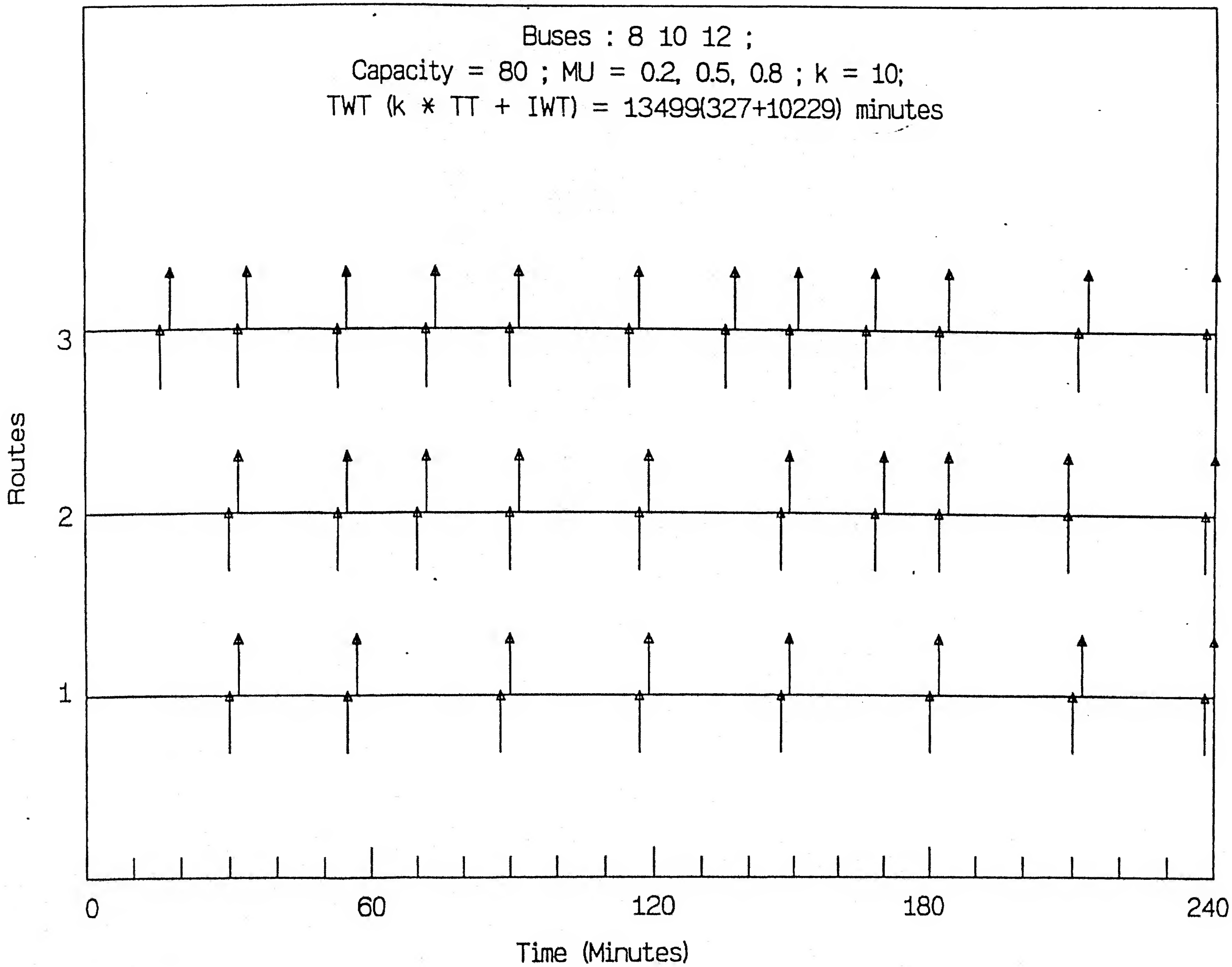

Figure 6.35: Optimal schedule for TWT consideration with bus capacity 80 and k = 10 (Non-uniform arrival pattern and unequal resources)

In this chapter, 33 schedules obtained using the proposed methodology are presented. A summary of the input variables for each of the 33 schedules are given in Table 6.1. It can be seen from the schedules that the proposed GA based procedure effectively determines optimal/near-optimal schedules for the capacity constrained transit scheduling problem.

F.No	CASE	No. of Buses	NTP	TP	C	μ	k	IWT (in minutes)	TT (in minutes)
6.1	1	10,10,10	980	120	40	UNI.	1	9948	280
6.2	2	10,10,10	970	0	35	0.2,0.2,0.2	1	9823	0
6.3	2	10,10,10	970	0	40	0.2,0.2,0.2	1	9387	0
6.4	2	10,10,10	970	0	50	0.2,0.2,0.2	1	9366	0
6.5	2	10,10,10	970	0	60	0.2,0.2,0.2	1	9371	0
6.6	2	10,10,10	970	0	80	0.2,0.2,0.2	1	9369	0
6.8	3	10,10,10	970	0	35	0.5,0.5,0.5	1	13280	0
6.9	3	10,10,10	970	0	40	0.5,0.5,0.5	1	10233	0
6.10	3	10,10,10	970	0	50	0.5,0.5,0.5	1	10223	0
6.11	3	10,10,10	970	0	60	0.5,0.5,0.5	1	10222	0
6.12	3	10,10,10	970	0	80	0.5,0.5,0.5	1	10224	0
6.14	4	10,10,10	1060	100	40	0.2,0.2,0.2	1	9532	322
6.15	4	10,10,10	1060	100	40	0.2,0.2,0.2	10	9793	225
6.16	4	10,10,10	1060	100	50	0.2,0.2,0.2	1	9458	229
6.17	4	10,10,10	1060	100	50	0.2,0.2,0.2	10	9440	215
6.18	5	10,10,10	1060	100	40	0.5,0.5,0.5	1	9526	940
6.19	5	10,10,10	1060	100	40	0.2,0.5,0.8	10	9934	656
6.20	5	10,10,10	1060	100	50	0.2,0.5,0.8	1	9496	351
6.21	5	10,10,10	1060	100	50	0.2,0.5,0.8	10	9726	225
6.22	6	10,10,10	1060	200	50	0.2,0.2,0.2	1	9610	558
6.23	6	10,10,10	1060	200	50	0.2,0.2,0.2	10	9694	531
6.24	6	10,10,10	1060	200	80	0.2,0.2,0.2	1	9519	414
6.25	6	10,10,10	1060	200	80	0.2,0.2,0.2	10	9821	376
6.26	7	10,10,10	1060	200	50	0.2,0.5,0.8	1	9729	1110
6.27	7	10,10,10	1060	200	50	0.2,0.5,0.8	10	9834	963
6.28	7	10,10,10	1060	200	80	0.2,0.5,0.8	1	9681	471
6.29	7	10,10,10	1060	200	80	0.2,0.5,0.8	10	9900	376
6.30	8	8,10,12	1050	100	50	0.2,0.2,0.2	1	10293	698
6.31	8	8,10,12	1050	100	50	0.2,0.2,0.2	10	10356	531
6.32	8	8,10,12	1050	100	80	0.2,0.2,0.2	1	9931	600
6.33	8	8,10,12	1050	100	80	0.2,0.2,0.2	10	10513	301
6.34	9	8,10,12	1050	100	80	0.2,0.5,0.8	1	9942	508
6.35	9	8,10,12	1050	100	80	0.2,0.5,0.8	10	10229	327

Chapter 7

Conclusions

In this study, the objective is to develop an algorithm for obtaining an optimal schedule for a transit system, under the assumption that vehicle capacity is finite i.e., bus capacity is limited and it may not be able to accommodate all the passengers waiting at the bus-stop.

The mathematical programming (MP) formulation and related complexities in solving the MP formulated scheduling problem was presented first, for infinite bus capacity (i.e., bus has sufficiently large capacity so that all passengers can be accommodated). Due to non-linear nature of the objective function and some constraints and presence of integer variables, it is very difficult to solve the problem by traditional methods. Next the MP formulation for Capacity - restrained scheduling problem is discussed and the complexities are presented.

In this work Genetic Algorithms an evolutionary optimization technique, was used to obtain the optimal schedule for the capacity constrained scheduling problem. Since, GAs allows external procedure based declarations, some constraints were taken care of through external procedures. The procedure based function declaration and coding of problem variables (i.e., eliminate those con-

straints which provide bounds on different variables) helped in reducing the complexity of the scheduling problem.

Results were obtained for a variety of problems and it has been found that GAs are able to find optimal schedules with reasonable computational resource. In the first type of problems the number of buses plying on each route is same. Optimal schedules were developed for different bus capacities and arrival pattern of passengers (Uniform and Non-uniform). In the second type of problems the number of buses plying on different routes are different. For these type of problems also optimal schedules were obtained for different bus capacities and arrival pattern of passengers. By studying these results it can be inferred that the procedure developed here does work effectively in determining optimal schedules for transit systems where demand at stops often exceeds the available bus capacity.

Although, this thesis shows that the procedure proposed here is able to find optimal schedules, no work has been done here to relate problem size to computation time of the algorithm. The author feels that this study is necessary and should be taken up at a later date.

Further, not much attention was given to efficient implementation (through computer programs) of the proposed algorithm. It is felt that the implementation can be made more efficient and this will further increase the speed at which the proposed algorithm determines the optimal schedule.

REFERENCES

[1] Chakroborty P., Deb K. and Subrahmanyam P.S., 1995."Optimal Scheduling of Urban Transit Systems using Genetic Algorithms," ASCE Journal of Trnsportation Engineering, Vol. 121(No. 6), pp. 544-553.

[2] Kikuchi S. and Parameswaran J., 1993. "Solving a Schedule Coordination Problem using Fuzzy Control Technique," Presented at Intelligent scheduling systems symposium at the ORSA/TIMS conference, San Francisco, CA.

[3] Srinivas B. 1995. "Network - Wide Transit System Scheduling using Genetic Algorithms," M.Tech thesis submitted to the department of civil engineering, IIT Kanpur.

[4] Reddy Y.S. 1996. "Transit System Scheduling under Stochastic Arrival Times," M.Tech thesis submitted to the department of civil engineering, IIT Kanpur.

[5] Subrahmanyam P.S. 1995. "Transit System Scheduling : Use of Genetic Algorithms," M.Tech thesis submitted to the department of civil engineering, IIT Kanpur.

[6] Glover F. 1975. "Improved Linear Integer Programming Formulation of Non-Linear Integer Problems," Management science, Vol. 22, pp. 455-460.

[7] Goldberg D.E. 1989. Genetic Algorithms in Search, Optimization and Machine learning. Mass., Addison-Wesley.

[8] Deb K. 1995. Optimization for Engineering Design. Printice Hall of India, New Delhi.